

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

ТИЖДЕНЬ НАУКИ-2020.
Факультет комп'ютерних наук і технологій

Збірник тез доповідей щорічної
науково-практичної конференції серед студентів,
викладачів, науковців, молодих учених і аспірантів

13–17 квітня 2020 року

Електронне видання комбінованого
використовування на DVD-ROM

м. Запоріжжя

УДК 004(06)
Т39

*Рекомендовано до видання Вченою радою
Національного університету «Запорізька політехніка»
(Протокол №4/20 від 02.03.2020 р.)*

Упорядник Голуб Т.В.

Редакційна колегія:

Наумик В. В., д-р техн. наук, професор (відпов. ред.)
Прушківський В. Г., д-р екон. наук, професор
Кузькін О.Ф., канд. техн. наук, доцент
Глушко В.І., канд. техн. наук, доцент
Климов О.В., канд. техн. наук, доцент
Антонов М.Л., канд. техн. наук, доцент
Савченко В.О., канд. техн. наук, доцент
Кабак В.С., канд. техн. наук, доцент
Касьян М.М., канд. техн. наук, доцент
Корольков В.В., канд. екон. наук, доцент
Дєдков М.В., канд. іст. наук, доцент
Васильєва О.О., канд. фіз.-мат. наук, доцент
Пуцина І.В., канд. пед. наук, доцент
Філей Ю.В., канд. юр. наук, доцент
Гайворонська Т.О., канд. філос. наук, доцент
Сажєєв В. М., канд. техн. наук, доцент
Висоцька Н. І., начальник патентно-інформаційного відділу

Тези доповідей друкуються методом прямого відтворення тексту, представленого авторами, які несуть відповідальність за його форму і зміст.

Т39 Тиждень науки-2020. Факультет комп'ютерних наук і технологій.
Тези доповідей науково-практичної конференції, Запоріжжя, 13–17 квітня 2020 р. [Електронний ресурс] / Редкол. :В. В. Наумик (відпов. ред.) Електрон. дані. – Запоріжжя : НУ«Запорізька політехніка», 2020. – 1 електрон. опт. диск (DVD-ROM); 12 см. – Назва з тит. екрана.
ISBN 978-617-529-264-8.

Зібрані тези доповідей, заслуханих на щорічній науково-практичній конференції серед студентів, викладачів, науковців, молодих учених і аспірантів. Збірка відображає широкий спектр тематики наукових досліджень, які проводяться на факультеті комп'ютерних наук і технологій Національного університету «Запорізька політехніка». Збірка розрахована на широкий загал дослідників та науковців.

ISBN 978-617-529-264-8.

© Національний університет
«Запорізька політехніка», 2020

ЗМІСТ

СЕКЦІЯ «КОМП'ЮТЕРНІ СИСТЕМИ ТА МЕРЕЖІ»	10
<i>Проскурін М.П.</i> Малорозрядні структури для паралельних обчислень.....	10
<i>Сирота А.В.</i> Загострення конкуренції на хмарному ринку	15
<i>Голуб Т.В., Зеленьова І.Я.</i> Підхід до апаратного прискорення процесу класифікації текстових документів.....	16
<i>Тягунова М.Ю.</i> Ефективність використання платформи ZOOM для дистанційної освіти	17
<i>Іванов М.А., Киричек Г.Г.</i> Реалізація програмного забезпечення для збору криптовалют.....	19
<i>Богатирьова Т.О., Зеленьова І.Я., Паромова Т.О.</i> Огляд засобів роботи з Big Data.....	22
<i>Верьовкін Д.С., Куликовська Н.А.</i> Порівняльний аналіз ігрових двигунів для настільних пристроїв	25
<i>Демченко М.М., Точилін С.Д.</i> Кросплатформний програмно-апаратний комплекс для електричних вимірів	26
<i>Діденко А.Є., Зеленьова І. Я., Дьячук Т.С.</i> Розробка суматору чисел з рухомою крапкою на FPGA.....	27
<i>Дьячук Т.С., Антонюк Д.С.</i> Захист Android-застосунку від злому	29
<i>Жигулін Д.І., Головіна К.П., Дьячук Т.С.</i> Використання багатофункціонального мережевого сніффера “Intercepter-NG” у відкритих Wi-fi мережах	31
<i>Загоренко І.О., Грушко С.С.</i> Система дистанційного керування на базі Arduino.....	33
<i>Здоровецький І.В., Куликовська Н.А.</i> Технології Big Data аналітики в розподілних системах обчислень	35
<i>Іващенко Н. А., Бредун А. Ю., Дьячук Т. С.</i> Комплекс моніторингу показників температури та вологості системи опалення та кондиціонування підприємства	36
<i>Іващенко Н.А., Федорова В.Ю., Дьячук Т. С.</i> Bullet Journal на платформі Android.....	38
<i>Касьян К.М., Островський Р.В.</i> Система компресії даних для Web.....	40
<i>Котенко А.О., Зеленьова І. Я., Голуб Т.В.</i> Розробка друкованої плати блоку управління в автоматизованій системі контролю рідини	41
<i>Куликовська Н.А., Азаров В.В.</i> Пошук запозиченої інформації в інтернеті	44
<i>Куликовська Н.А., Азаров В.В.</i> Аналіз якості тексту	45

<i>Лубянов Д. В., Дьячук Т. С.</i> Програмування пристроїв дистанційного керування на Arduino	46
<i>Луценко Н.В., Паромова Т.О., Бережко Я.А., Дмитренко К.О.</i> Використання системи JIRA для тестування та керування проектами	49
<i>Луценко Н.В., Паромова Т.О., Іващенко Н.А., Бредун А.Ю.</i> Порівняльна характеристика мов програмування C++ та Clojure	52
<i>Луценко Н.В., Паромова Т.О., Костецький Д.В.</i> Чат-боти у сучасному ІТ-просторі	55
<i>Луценко Н.В., Паромова Т.О., Поліщук В.О.</i> Використання системи Docker при розробці програмного забезпечення	58
<i>Назаренко О.І., Грушко С.С.</i> Система вимірювання атмосферних показників в концепції інтернета речей	60
<i>Наливайко А.А., Тягунова М.Ю.</i> Методика проектування графічного інтерфейсу мобільного додатка.....	62
<i>Нестюрич С.І., Точилін С.Д.</i> Web-додаток - лабораторна робота «Визначення коефіцієнта в'язкості рідини»	64
<i>Предко В.Ю., Киричек Г.Г.</i> Мікросервіси проти монолітів	66
<i>Предко В.Ю., Польська О.В.</i> Роль тайм-менеджменту в організації ІТ-проекту	69
<i>Сгадов С. О., Тур'янський Е.Д.</i> Об'єктно-орієнтований підхід до дизайну інтерфейсів мікропроцесорних пристроїв.....	70
<i>Стусь С.В., Тягунова М.Ю.</i> Актуальність сучасних CMS систем	72
<i>Твердохліб М.Ю., Киричек Г.Г.</i> Система з підтримкою терміналу обслуговування	76
<i>Тіменко А.В., Бубенець О.В.</i> Використання нейромережевої моделі для ідентифікації даних за рентгенівськими знімками	78
<i>Тіменко А.В., Вусу-Гову С.Е.</i> Мікросервісна архітектура як спосіб розширення та вдосконалення монолітного додатку для письменників	82
<i>Тіменко А.В., Дринов Д.О.</i> CMS Wordpress та PHP Framework Laravel, та сфери їх використання	84
<i>Тіменко А.В., Опара О.С.</i> Нейронні мережі у сучасному світі	86
<i>Точилін С.Д., Демченко М.М., Точилін Д.С.</i> Кросплатформний віртуальний низькочастотний генератор	90
<i>Точилін С.Д., Нестюрич С.І., Точилін Д.С.</i> Комп'ютерна програма для оцінки вартості об'єктів нерухомості на основі нейронної мережі.....	91
<i>Точилін С.Д., Демченко М.М., Точилін Д.С.</i> Кросплатформний віртуальний осцилограф.....	92
<i>Тягунова М.Ю., Кошарна К.О.</i> Використання технології RFID для складського обліку	94
<i>Тягунова М.Ю., Литвинов А.С.</i> Використання мобільної дослідної станції ICLEVO KOBUKI у навчанні.....	95

<i>Тягунова М.Ю., Циб О.О.</i> Програмні засоби для роботи з платами сімейства Arduino	96
<i>Тягунова М.Ю., Чумак І.Т.</i> дослідження мобільної роботизованої платформи Irobot Create 2	98
<i>Чубіч А.І., Киричек Г.Г.</i> Впровадження KANBAN-дошки при організації розробки програмного забезпечення	99
<i>Шавло Г.В., Киричек Г.Г.</i> Використання веб технологій в системах віртуальної комерції	102
<i>Тягунова М.Ю., Крупій О.С.</i> Онлайн-сервіс вивчення іноземних мов.....	106
<i>Щербак С.О., Скрупський С.Ю., Щербак Н.В.</i> Підхід до спрощення процесу розробки програмного забезпечення	107
<i>Кармелюк К.О., Кудерметов Р.К., Польська О.В.</i> Анімація руху моделі супутника із Qt Quick 3D API.....	109
СЕКЦІЯ «ПРОГРАМНІ ЗАСОБИ».....	111
<i>Субботін С.О., Литвин В.А.</i> Аналіз методів цифрової обробки сигналів для побудови діагностичних моделей	111
<i>Субботин С.А., Гофман Е.А., Корниенко А.В.</i> Построение диагностических моделей с предварительной цифровой обработкой сигналов	113
<i>Соколянський О.В., Пархоменко А.В.</i> Дослідження та використання сучасних фреймворків для модернізації віддаленої лабораторії.....	116
<i>Туленков А.В., Пархоменко А.В.</i> Забезпечення нормального теплового режиму роботи електрощитового обладнання системи домашньої автоматизації.....	117
<i>Федорончак Т.В.</i> Методи оцінювання складності задач в Agile методологіях	118
<i>Колпакова Т.О.</i> Особливості підбору команди для розробки WEB-проектів за методологією Agile.....	119
<i>Леоценко С.Д., Олійник А.О., Субботін С.О.</i> Використання методу котячих зграй як ройової процедури для синтезу нейронних мереж	121
<i>Вічева Г.Ю., Зайко Т.А.</i> Стенографічні методи як засіб боротьби з несанкціонованим доступом.....	123
<i>Бережний О.Ю., Зайко Т.А.</i> Методи антивідладжувального програмування	125
<i>Зиранов Ю. О., Зайко Т. А.</i> Принцип боротьби з піратством всучасній індустрії відеоігор.....	127
<i>Саман М.В., Зайко Т.А.</i> Використання ТВВЗ-технології в розробці програмного забезпечення.....	129
<i>Куковинець О.Є., Зайко Т.А.</i> Методи забезпечення безпеки автоматизованих систем обробки інформації в банках	131

<i>Селетков Є. А., Зайко Т. А.</i> Інформаційний захист в індустрії відеоігор .	133
<i>Зайко Т.А., Козлов В.В.</i> Обфускація як засіб захисту вихідного коду програми	135
<i>Марков О. С., Зайко Т. А.</i> Реалізація ускладнення логіки як метод захисту від дизасемблювання	137
<i>Марічев Д.О., Зайко Т.А.</i> Принцип роботи SSL сертифікату шифрування в HTTPS протоколі.....	139
<i>Сущевських О. В., Зайко Т. А.</i> Сучасні технології дам্পінга і захисту від нього	141
<i>Зосенко К. В., Зайко Т. А.</i> Інструменти злому програмного забезпечення	143
<i>Кузнецов Т.В., Зайко Т. А.</i> Розробка і реалізація вбудованого захисту програм	145
<i>Дьячков В.В., Зайко Т. А.</i> Безпека Веб-додатків. Міжсайтовий скриптинг (XSS) та методи боротьби з ним	147
<i>Абдуллаєв М. С., Зайко Т. А.</i> Захист від несанкціонованого копіювання .	149
<i>Архипов А.С., Туленков А.В., Пархоменко А.В.</i> Розробка програмно-апаратного комплексу для контролю показників експлуатації обладнання домашньої автоматизації.....	150
<i>Задоян М.К., Соколянський О.В., Пархоменко А.В.</i> Розробка мобільного інтерфейсу для віддаленого керування гібридною лабораторією.....	152
<i>Козлов В.В., Туленков А.В., Пархоменко А.В.</i> Використання датчика MAX30102 для проектування парамедичного браслету	154
<i>Куц А.В., Пархоменко А.В., Гладкова О.М.</i> Оцінка якості відео-контенту для медичної галузі на платформі YOUTUBE.....	156
<i>Малюков М.К., Пархоменко А.В.</i> Впровадження технологій віртуальної реальності для популяризації сучасних навчальних лабораторій.....	157
<i>Яремченко Є.В., Туленков А.В., Пархоменко А.В.</i> Розробка голосового сповіщення системи домашньої автоматизації для користувачів з особливими потребами	158
<i>Білов О.І., Залюбовський Я.І., Пархоменко А.В., Гладкова О.М.</i> Віртуальні середовища для дослідження систем розумний будинок	160
<i>Захряпа Р.А., Кривицун О.В.</i> Програмна реалізація гри з використанням технології Unreal Engine.....	161
<i>Сердюк С.М., Камінська Ж.К., Шутко А.С.</i> Розробка програми нотаток «ANOTES»	163
<i>Сердюк С.М., Камінська Ж.К., Заїкіна К.Є.</i> Розробка концепту «Розумна шафа».....	165
<i>Сердюк С.М., Камінська Ж.К., Бірюк В.О., Хижняк Д.В.</i> Інтерфейс системи електронного меню	167
<i>Сердюк С.М., Камінська Ж.К., Клоков А. О., Петелін Д. Д.</i> Проблемно-орієнтований інтерфейс багатофункціонального бортового комп'ютера.	169

<i>Сердюк С.М., Камінська Ж.К., Сокол Р.В.</i> Розробка інтерфейсу музичного плеєра для гітаристів	171
<i>Сердюк С.М., Камінська Ж.К., Косенков І.С.</i> Проблемно-орієнтований інтерфейс розумних вагів	173
<i>Сердюк С.М., Камінська Ж.К., Терлецький С.В., Денисенко С.В., Ліпихін С.Б.</i> Розробка концепту «Розумний будинок»	175
<i>Сердюк С.М., Камінська Ж.К., Фоміна М.О.</i> Розробка проблемно – орієнтованого інтерфейсу розумного холодильника	177
<i>Сердюк С.М.І, Камінська Ж.К., Колесникова М.В., Дем'янова Д.К.</i> Проблемно-орієнтований інтерфейс застосунку “Безпілотне таксі”	179
<i>Сердюк С.М., Камінська Ж.К., Чабан І.С.</i> Людино-орієнтований підхід до проектування інтерфейсу системи керування світлофорами	181
<i>Гутаров А.О., Каплієнко Т.І.</i> Програмна реалізація адміністративної частини вебсайту «Інформаційний блог університету»	183
<i>Кругов В.В., Каплієнко Т.І.</i> Програмна реалізація забезпечення для проведення турнірів з фехтування	185
<i>Нестеренко М.А., Каплієнко Т.І.</i> Розробка клієнт-серверного застосунку для обліку часу співробітників з гнучким графіком роботи	187
<i>Петренський В.О., Каплієнко Т.І.</i> Програмна реалізація клієнтської частини вебсайту «Інформаційний блог університету»	189
<i>Каплієнко Т.І., Шлома Р.О.</i> Розробка інформаційно-комунікаційної технології для автоматизації роботи з контингентом студентів	191
<i>Щетинін М.О., Каплієнко Т.І.</i> Програмна реалізація підсистеми ведення електронних відомостей для вебсайту «Інформаційний блог університету»	193
<i>Каплієнко Т.І., Галевська І.Д.</i> Програмна реалізація вебсайту для прослуховування музики з взаємодією з користувачем	195
<i>Льовкін В.М., Мироненко Н.В.</i> Розробка програмного забезпечення керування роботою студентського гуртожитку	197
<i>Льовкін В.М., Мяло О.Є.</i> Розроблення програмного забезпечення графічного редактора для візуалізації концепцій	199
<i>Льовкін В.М., Чорнобель Д.С.</i> Програмна реалізація Telegram-бота для підтримки студентів факультету комп'ютерних наук	201
<i>Льовкін В.М., Ярмачек М.В.</i> Розробка мобільної 3D-гри за допомогою середовища UNITY	203
<i>Poliakov O.</i> Visualization of algorithms using remote laboratories	205
<i>Дейнега Л.Ю., Басанець М.І., Діденко А.Є., Камєнський Д.В.</i> Складання розкладу робіт для задач з функціями штрафів	206
<i>Костюк П.М., Федорончак Т.В.</i> Порівняння архітектур SPA та MPA в реалізації веб застосунків	208

<i>Олійник А.О., Федорченко Є. М., Злиденний А.О.</i> Програмна реалізація аркадного застосунку для мобільних пристроїв	210
<i>Олійник А.О., Федорченко Є. М., Харченко А.С.</i> Розробка модифікованого генетичного методу для вирішення задачі розміщення об'єктів електропостачання	211
<i>Степаненко О.О., Федорченко Є. М., Трошина Н.Є., Гончаренко Д.А., Бородин О.О.</i> Програмна реалізація системи моніторингу стану комп'ютера	213
<i>Степаненко О.О., Федорченко Є. М., Трошина Н.Є., Гончаренко Д.А., Брильов Є.С.</i> Програмна реалізація Веб-сайту ресторану	214
<i>Степаненко О.О., Федорченко Є. М., Трошина Н.Є., Гончаренко Д.А., Шлома Р.О.</i> Розробка бази даних для навчального відділу та програмна реалізація модуля генерації плану навчання	216
<i>Качан О.І., Федорченко Є. М., Степаненко О.О., Скачко Л.П., Трошина Н.Є., Гончаренко Д.А., Коломоєць В.М.</i> Програмна реалізація картографічного модуля для тренажера артилерії з використанням технології доповненої реальності.....	218
<i>Миронова Н.А., Ліпихін С.О., Терлецький С.В., Денисенко С.В.</i> Сучасні технології контейнеризації.....	219
<i>Миронова Н.О., Колесникова М.В., Дем'янова Д.К., Сокол Р.В.</i> Розробка кроссплатформного мобільного застосунку для відображення погоди на Flutter	221
<i>Миронова Н.О., Третьяков Д.А., Сидорський В.С., Радченко І. А.</i> Розробка кроссплатформного мобільного застосунку «Менеджер витрат» на Flutter.....	223
<i>Миронова Н.О., Гальченко В.В., Перетятко В.В., Савенко Х.С.</i> Розробка кроссплатформного мобільного застосунку «WorldTime» на Flutter	225
<i>Миронова Н.О., Ковальов Р.В., Резніченко А.С., Рогова Є.В., Тихоновська Т.В.</i> Візуалізація криптографічних алгоритмів для Blockchain технології. RSA (Rivest–Shamir–Adleman)	227
<i>Миронова Н.О., Косенко Є.А.</i> Розробка SPA Застосунків за допомогою javascript фреймворку VUE.JS.....	229
<i>Миронова Н.О., Сташук Д.А., Лактіонов В.В., Павелко О.О.</i> Програмна реалізація візуалізації алгоритмів сортування	231
<i>Миронова Н.О., Гальченко В.В.</i> Візуалізація алгоритмів сортування	233
<i>Миронова Н.О., Подиряко Б.К.</i> Розробка мобільного застосунку для перегляду та аналізу даних матчів та статистики гравців багатокористувацької гри	235
<i>Миронова Н.О., Петухов М. М.</i> Розробка мобільного застосунку для планування та обліку часу, витрат та завдань	237

<i>Миронова Н.О., Куделя Д.С.</i> Розробка програмної системи матчмейкінгу для багатокористувацьких онлайн відеоігор	239
<i>Миронова Н.О., Колпакова Д.В.</i> Розробка мобільного застосунку для інвентаризації товарно-матеріальних цінностей.....	241
<i>Миронова Н.О., Сизоненко Б.В.</i> Розробка гри з використанням технологій UNITY для мобільних пристроїв	244
<i>Миронова Н.О., Волков П.С.</i> Програмна реалізація засобів синтаксичного аналізу зовнішніх даних.....	246
<i>Миронова Н.О., Нездропка І.П.</i> Розробка мобільної карти пунктів безкоштовної видачі води.....	248
СЕКЦІЯ «СИСТЕМНИЙ АНАЛІЗ ТА ОБЧИСЛЮВАЛЬНА МАТЕМАТИКА»	250
<i>Бахрушин В.Є.</i> Особливості статистичних даних для прийняття рішень щодо запобігання поширенню Covid-19.....	250
<i>Корніч Г.В., Широкопад Д.В.</i> Еволюція янусоподібних Ni-Al кластерів під впливом низькоенергетичних кластерів Ag та Ag ₁₃	252
<i>Бакурова А.В., Савранська А.В., Шевчук М.В.</i> Оптимізація портфелю сезонних товарів	254
<i>Кривцун О.В.</i> Допустимі графи та їхні необхідні та достатні гіперкуби ..	257
<i>Денисенко О.І.</i> Чисельне моделювання нестационарного процесу вимивання домішок з водоймищ.....	258
<i>Терещенко Е.В., Литвиненко А.В.</i> Задача про спорідненість ліній соняшнику	259
<i>Подковаліхіна О.О., Січкара М.О.</i> Аналіз розв'язку задачі завантаження з нормальними параметрами	260
<i>Білий В.В.</i> Система управління взаємодією з клієнтами стоматології.....	261

СЕКЦІЯ «КОМП'ЮТЕРНІ СИСТЕМИ ТА МЕРЕЖІ»

УДК 681.3.01; 004.272.3

Проскурін М.П.

канд. техн. наук, доц. НУ "Запорізька політехніка"

МАЛОРОЗРЯДНІ СТРУКТУРИ ДЛЯ ПАРАЛЛЕЛЬНИХ ОБЧИСЛЕНЬ

Загальновідомо, що максимальну швидкодію забезпечують обчислювальні системи з апаратною реалізацією алгоритму. В даний час для побудови обчислювальних систем такого класу використовуються програмовані інтегральні схеми (ПЛІС). У роботі [1] відзначається, що для досягнення найбільшої ефективності і продуктивності при вирішенні сильно пов'язаних завдань необхідно використовувати безліч ПЛІС, об'єднаних за допомогою високошвидкісних інтерфейсів в єдине обчислювальне поле. Це дозволяє структурно реалізувати базовий підграф розв'язуваної задачі і суттєво, на один...два порядки, підвищити швидкість виконання завдання в цілому. Описана в роботі концепція побудови реконфігурованих обчислювальних систем (РОС) на основі ПЛІС дозволила створити ряд систем, які за реальною продуктивністю в одиниці об'єму та енергоефективності істотно випереджають як кластерні обчислювальні системи, так і відомі вітчизняні та зарубіжні РВС. В якості елементної бази для побудови РОС використані ПЛІС великого ступеня інтеграції фірми Xilinx сімейства Virtex, з'єднані в єдиний обчислювальний ресурс швидкісними каналами передачі даних (0,8 ГГц) [1].

Відомий інший шлях, що веде до апаратної реалізації алгоритму: використання однорідних обчислювальних середовищ (ООС), в яких має місце так званий структурний спосіб організації обчислювального процесу, наприклад на основі одно-, двох- розрядних процесорів (в загальному підході – мало-розрядний процесор). Прикладом втілення їх в ООС є використання матричних обчислювальних системи (МОС), які пристосовані для вирішення завдань, що характеризуються паралелізмом незалежних даних. Такі системи мають спільний керуючий пристрій, що генерує потік команд, і велике число процесорних елементів (ПЕ), що працюють паралельно і обробляють (кожен з них) свій потік даних [2]. В основі ПЕ лежить матричний процесор (array processor), що складається з масиву ПЕ. Однак на практиці, щоб забезпечити достатню ефективність системи (при вирішенні широкого кола завдань), необхідно організувати зв'язки між ПЕ так, щоб: а - організувати функціонал (налаштування на і-те завдання); б - забезпечити режим повного завантаження кожного ПЕ роботою. Саме характер зв'язків між ПЕ і визначає властивості МОС. Подібна схема може бути застосована і для векторних

обчислень. Матричний процесор інтегрує багато ідентичних функціональних блоків (ФБ), логічно об'єднаних в матрицю, які працюють в SIMD- стилі. В більшості матричних SIMD-систем в якості процесорних елементів застосовуються прості RISC-процесори з локальною пам'яттю обмеженою ємності. Наприклад, кожен ПЕ системи MasPar MP-1 складається з 4-розрядного процесора з пам'яттю ємністю 64 Кбайт. В системі MPP використовуються однорозрядні ПЕ з пам'яттю 1 Кбіт кожен, а в CM-2 ПЕ являє собою однорозрядних процесор з 64 Кбіт локальної пам'яті. Завдяки простоті ПЕ масив може бути реалізований у вигляді однієї надвеликої інтегральної мікросхеми (HBIC), що дозволяє скоротити число зв'язків між ІС і, отже, габарити. Так, одна HBIC в системі CM-2 містить 16 процесорів (без блоків пам'яті), а в системі MasPar MP-1 HBIC складається з 32 процесорів (також без блоків пам'яті) [2].

Використання кремнієвої (Si) пластини (як конструктивної одиниці) дозволяє обійтися проектними нормами добре обкатаного рівня технологій таким чином, щоб питома щільність транзисторів в обчислювальній системі, реалізованої на пластинах, буде в підсумку на один-два порядки вище, ніж в апаратурі, побудованої на мікросхемах (навіть якщо вони будуть виготовлятися по гранично допустимим на сьогоднішній день технологічним нормам). Саме готовність ООС «перейти на пластину» - основна перевага МОС перед іншими напрямками в обчислювальній техніці - мікропроцесорними і трансп'ютерном системами, систолойкою, програмованими гратчастими полями (FPGA) і логічними інтегральними схемами (ПЛІС), ін.

Але щоб «перейти на пластини», потрібні ООС нового покоління - наприклад такі, що отримали назву пульсуючі інформаційні решітки (пульсири) [3], бо ця термінологія більш точно відображає процес обробки інформації в подібних регулярних структурах. У пульсирі першорядну увагу приділено питанням забезпечення надійності та живучості, що, в свою чергу, гарантує технологічність виробництва і ефективність експлуатації МОС.

На рис. 1а,б показана частина 2D (двовимірної) матриці пульсирів, комірки якої з'єднані між собою локальними і глобальними інформаційними шинами. Як видно з рис. 1а, кожна комірка (мінімальний елемент матриці пульсира) виконує функції над восьмирозрядними операндами і може виступати, як ПЕ (що має малорозрядне АЛУ з відповідною периферією) і/або допоміжні пристрої, що їх оточують: блоки вводу/виводу, регістри, п'ять кросконекторів сигналів, ін. та може бути з'єднана ближніми зв'язками з вісьмома комірками - сусідніми пульсирами. Ці блоки вводу / виводу мають фіксовану комутацію та зображені на рис.1а тонкими лініями. Для зв'язку з дальніми комірками (від обраної) можуть бути застосовані інші 8-ми розрядні шини, що показані на рис.1б, які можуть функціонувати активуванням перетину ліній «координатної сітки» (режим типу запис/зчитання): напри-

клад, комірка ij (ПЕ ij зліва внизу) передає дані до комірки mn (ПЕ mn , справа уверху) або іншим способом. Блоки 2D пульсирів, які розташовані на обоїх сто-ронах пластини, можуть складатися із кількох сотен (тисяч) ПЕ, їх допоміжних схем та з'єднуватися між собою електричними (рис.1а,б) і оптичними каналами, як зображено на рис.2а [3].

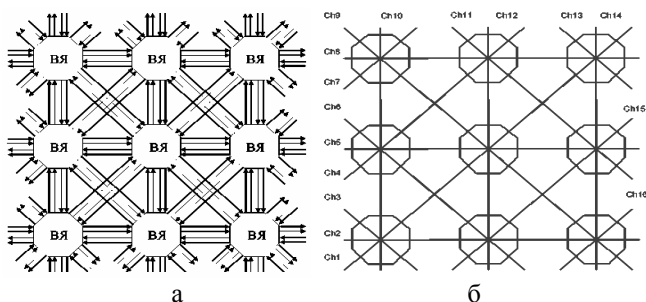
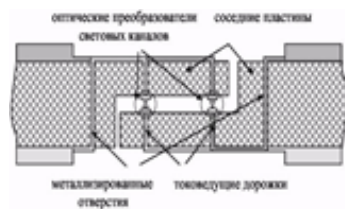


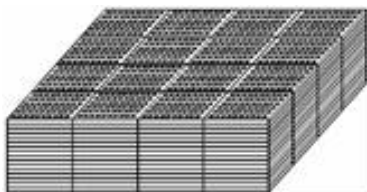
Рисунок 1 - Ескіз частини блоку 2D пульсира і його ближніх електричних зв'язків (а) та 2D матриця пульсира, напрями його можливих (активованих чи ні) дальніх електричних зв'язків (б)

Кожен з оптичних каналів рис.2а складається з пари перетворювачів - електронно-оптичного і оптоелектронного (дуплексний комутатор). Кожен «сполучений» комутатор i -го блоку може здійснювати передачу/прийом даних в напрямку до та від j -го сусіднього блоку. Використання «сполучених» комутаторів дозволяє здійснити передачу даних між «пов'язаними» комірками і транзит даних через усю пластину, які надходять світловими потоками зверху і знизу, при цьому виходи «пов'язаних» кліток і виходи перетворювачів оптичних каналів є входами відповідного «сполученого» комутатора. Кому-тація входів «сполучених» комутаторів задається безпосередньо кодом управління відповідної «пов'язаної» комірки. Інші два входи «сполучених» оптоелектронних комутаторів, що розташовані на верхній і нижній поверхнях пластини, є каналами прийому даних локальних шин. Оптоелектронний ко-мутатор складається з чотирьох каналів. Аналогічно влаштовані комутатори для нижньої поверхні пластини з Si [3].

Кілька паралельно розташованих 2D блоків (нижні, верхні від обраного) формують 3D структуру (рис. 2б), де електричний (або оптичний) тип зв'язку між ними виконує комутаційну функцію між кількома десятками 2D шарів, які виконані на основі однорідних структур на Si пластині [3].



а



б

Рисунок 2 - Загальний вигляд: оптичних зв'язків між сусідніми 2D блоками (а) та варіант реалізації об'ємної 3D структури із кількох десятків плоских 2D структур (б)

Подібні особливості мають конструктивні технологічні рішення (КТР) оптоелектронних цифрових ІС (ОЦІС), що зображені на рис.3а,б [4,5] і які раніше запатентовані автором цих тез. В основі їх побудови використано схему оптоелектронного логічного елемента nАБО-НІ (базис Пірса), що реалізована на мікропотужних оптронах UVЧ діапазону. Саме їх можливо обрати у якості оптоелектронного базису для створення комірок одно-, двох-малорозрядних процесорів (типу пульсирів), але всі вхідні, вихідні, управляючі, інформаційні, інші зв'язки в них мають виключно оптичний характер.

Висновок. Наведені КТР пульсирів, ОЦІС, їх комбінацій можуть бути взяті за основу для створення ООС на основі малорозрядних обчислювальних комірок з електричними і/або оптичними зв'язками між ними. Використання пульсирів і/або ОЦІС (на базі КТР типу 2D, 3D рис.1а,б, рис.2а,б) при створенні ООС відкриває нові можливості їх застосування у якості основи для моделювання РОС, ООС, МОС і схем на їх базі та суттєвого підвищення продуктивності паралельних обчислювань (наприклад, отримання скалярних добутків С при множенні операндів $A \times B$: схеми Дадда, Уоллеса, ін.).

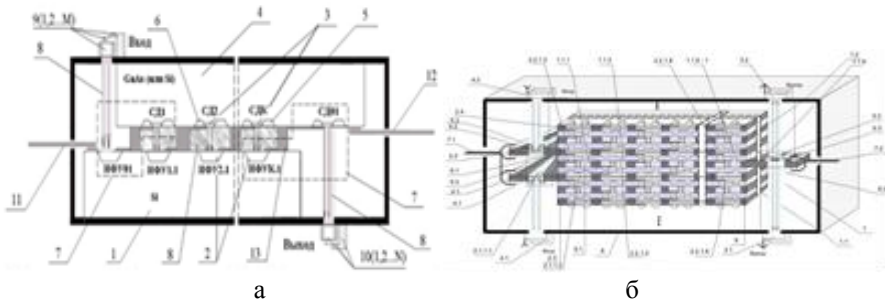


Рисунок 3 - Ескізи ОЦІС і їх вигляд: торець блоку 2D схеми ОЦІС (оптоелектронні логічні елементи пАБО-НІ позначені пунктиром) (а); 3D схема ОЦІС має 7 шарів: 3(нижні) + 1 (середина) + 3 (верхні). В обох ОЦІС вхідні (зліва), внутрішні, вихідні (справа) зв'язки є оптичними (б)

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Реконфигурируемые мультиконвейерные вычислительные структуры / И.А. Каляев, И.И. Левин, Е.А. Семерников, В.И. Шмойлов. Изд. 2-е, перераб. и доп.. Под ред. И.А. Каляева. Ростов-на-Дону: Изд-во ЮНЦ РАН, 2009. – 344 с.
2. Новосибирский государственный университет. Факультет информационных технологий. Матричные вычислительные системы; Матричный процессор [Электронный ресурс]. – Режим доступа: www.nsc.ru/win/elbib/data/show_page.dhtml (дата звернення: 05.03.2020). – Назва з екрана.
3. Гузик В.Ф. Пульсирующие информационные решетки с матричной коммутацией / В.Ф. Гузик, В.И. Шмойлов, Г.А. Кириченко // Известия ВУЗов. Северо-Кавказский регион. Технические науки. – 2014. – № 6. – С.3-11.
6. Пат. на корисну модель 10133 Україна, МПК G02F 3/00. Цифровий пристрій з оптоелектронним блоком / Проскурін М.П., Білявська О.С., Демиденко О.О. (Україна) - № у 2005 00239; Заявл.11.01.2005; опубл. 15.11.2005, Бюл. №11. – 4 с.
7. Пат. на корисну модель 51219 Україна, МПК G02F 3/00. Оптоелектронна інтегральна схема для цифрових автоматів / Проскурін М.П. (Україна) - № у 2009 13491; Заявл.24.12.2009; опубл. 12.07.2010, Бюл. №13. – 5 с.

ЗАГОСТРЕННЯ КОНКУРЕНЦІ НА ХМАРНОМУ РИНКУ

Конкуренція між учасниками хмарного ринку не припиняється ні на мить.

При цьому використовуються усі доступні інструменти і ресурси компаній.

Інакше, той, хто не веде конкурентну боротьбу і не змагається з конкурентами, втратить частину ринку і більше може туди не повернутися.

Попри те, що хмарні технології не так давно впровадилися в сучасні інформаційно-обчислювальні системи, що ведуть фірми хмарних сервісів наперебій почали пропонувати свої послуги.

Відчувши фінансовий інтерес на хмарному ринку, сталося загострення конкурентної боротьби серед ІТ магнатів за сфери впливу.

Так, щоб зміцнити свої позиції американський виробник серверів і комп'ютерів Dell об'єднався з EMC -найпотужнішим розробником систем зберігання даних, сума угоди склала \$67 млрд.[1]

Таким чином Dell постарається витіснити таких виробників як Microsoft, Amazon і Google з ринку хмарних технологій.[2]

З іншого боку хмарному підрозділу Google, що продає хмарні сервіси, керівництво компанії Alphabet може позбавити фінансування, якщо до 2023 р. не вдасться обійти конкурентів, Amazon і Microsoft.

Для ліквідації розриву з конкурентами Google, придбає одного з провідних виробників програмного забезпечення в хмарній сфері компанію Arigee за \$625 млн.[2]

До сфери хмарних сервісів підключився і Пентагон надавши багатомільярдний контракт компанії Microsoft.

Перерозподіл ринку хмарних послуг диктує нові умови.

Так, здавалось, нещодавно ще міцні союзи розпадаються, навпроти колишні непримиренні суперники об'єднуються, керуючись фінансовими інтересами.

Результатом таких дій, сталося об'єднання Microsoft і Oracle, VMware з AT&T та ін. [3].

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Сайт Internetua [Електроний ресурс]. – Режим доступу: <https://internetua.com/google-planiruet-za-pyat-let-likvidirovat-otstavanie-ot-liderov-oblacsno-go-rynka>

2. Сайт Quote [Електроний ресурс]. – Режим доступу: <https://quote.rbc.ru/news/article/5de04f429a79470598470830>

3. Сайт De-novo [Електроний ресурс]. – Режим доступу: <https://www.de-novo.biz/ru/blog/rinok-hmar-v-ukrayini-2019-maksim-agyeyev-ceo-de-novo-70>

УДК 004.4

Голуб Т.В.¹, Зеленьова І.Я.²

¹асист. НУ «Запорізька політехніка»

²канд. техн. наук, доц. НУ «Запорізька політехніка»

ПІДХІД ДО АПАРАТНОГО ПРИСКОРЕННЯ ПРОЦЕСУ КЛАСИФІКАЦІЇ ТЕКСТОВИХ ДОКУМЕНТІВ

Класифікація текстових документів в одну із заздалегідь визначених категорій є складним процесом. І, як будь-який складний процес, вимагає значних часових витрат на його реалізацію.

В загальному сенсі класифікація складається з двох значних етапів: визначення ознак категорій і безпосередньої класифікації певного тексту в одну з категорій.

Визначення ознак категорій виконується на основі опорної вибірки текстів, категорії приналежності яких визначені заздалегідь експертами. Даний процес супроводжується великим обсягом розрахунків і порівнянь, в результаті чого вимагає значних часових витрат. При цьому він виконується в більшості випадків один раз. Тому вдосконалення даного процесу за критерієм часу не є доцільним.

Безпосередня класифікація певного тексту в одну із заздалегідь визначених категорій виконується на основі порівняння ознак цього тексту з ознаками кожної категорії. Таке порівняння також вимагає значних часових витрат, пов'язаних з великою кількістю ознак. При цьому даний процес повторюється багаторазово для кожного документу, що аналізується. Тому його оптимізація за часом є актуальною.

На даний момент класифікатори текстових документів, на які покладено виконання другого етапу, реалізовано в більшості випадків в програмному варіанті. Це супроводжується складністю прискорення даного процесу, зумовленого підвищенням вимог до розрахункових потужностей обчислювальних приладів, що виконують відповідні функції.

Побудова ж класифікатора на основі програмно-апаратної реалізації дозволить використати внутрішні ресурси, які надають можливість паралельного виконання процесів без необхідності збільшення розрахункових потужностей приладу.

Програмна частина програмно-апаратної реалізації виконує етап формування простору ознак категорій, а на апаратну частину покладено етап класифікації тексту. При чому останній включає крок попередньої підготовки тексту, який полягає у виділенні ознак документу, що аналізується, і крок безпосередньої класифікації тексту.

Таким чином, перенесення обох кроків етапу класифікації на апаратну частину дозволяє створити автономний прилад для класифікації текстових документів, перелік ознак категорій якого формується програмною частиною.

УДК 004.514

Тягунова М.Ю

канд. техн. наук, доц. НУ «Запорізька політехніка»

ЕФЕКТИВНІСТЬ ВИКОРИСТАННЯ ПЛАТФОРМИ ZOOM ДЛЯ ДИСТАНЦІЙНОЇ ОСВІТИ

На сьогодні, при всесвітньому карантині [1], обґрунтування актуальності дистанційної освіти, на мій погляд, є зайвим. І в цих нелегких економічних, психологічних та побутових умовах дуже важливим є вміння швидко адаптуватися до зовнішніх умов.

Існує безліч різних платформ та додатків для реалізації дистанційної освіти, наприклад, Zoom [2], Google Classroom [3], Skype[4], Webinar [5], ClickMeeting [6] та багато інших. Проте, я вважаю доцільним використовувати для спілкування зі студентами ВНЗ саме платформу Zoom.

Із власного досвіду використання даної платформи можу відокремити наступні основні її переваги:

1. Можливість використовувати платформу безкоштовно з описаними нижче перевагами. Це один із перших критеріїв, який впливає на вибір програмного засобу, особливо, в умовах економічної кризи;

2. Можливість чути та бачити багатьох студентів одночасно. Це, безспірно, максимально наближує відеозустріч до умов проведення лекції у межах ВНЗ;

3. Можливість студентам задавати питання під час лекції, сигналізуючи про це лектора зручним графічним інструментом, наданим Zoom. Завдяки цьому студент має можливість бути максимально залученим до активної діяльності під час лекції, що дуже позитивно впливає на засвоєння матеріалу;

4. Можливість вести запис відеолекції вбудованими засобами Zoom. Це особливо важливо у двох аспектах: для викладача існує можливість проаналізувати активність студентів та власний рівень викладання для

подальших висновків, а студент має можливість отримати відеозапис від викладача та продивитись його офлайн. Особливо це актуально для студентів, що хворіють під час проведення лекції, або взагалі не мають технічної можливості приймати участь в дистанційній онлайн освіті. Також це вагома допомога під час підготовки до іспиту, або контрольної роботи;

5. Наявність чату. Дозволяє студентам, не перебиваючи лектора, задавати власні питання, або залишати коментарі, а ще допомагає вирішити технічні питання, якщо студент не чує та/або не бачить викладача з технічних причин.

6. Можливість у реальному часі демонструвати екран власного комп'ютера, або презентації, або будь-якого іншого відео чи графічного матеріалу, не втрачаючи візуального контакту зі студентами. Це, по-перше, дозволяє більш якісно викладачеві надати лекційний матеріал, а студенту, відповідно, засвоїти; по-друге, надає можливість більш якісного пояснення матеріалу, якщо виникли питання, бо існує можливість для цього використовувати будь-які інші ресурси, не заплановані заздалегідь, та одразу демонструвати студентові.

7. Можливість студентам зручними вбудованими засобами давати оцінку заданим питанням та емоційно реагувати на поданий матеріал. Це також, звичайно, впливає на якість освіти.

8. Доступність та швидкість з точки зору опанування платформи Zoom викладачем та простота і зрозумілість інтерфейсу як для викладача, так і для студента.

Таким чином, платформа Zoom дозволяє викладачам надавати, а студентам отримувати, високоякісну дистанційну освіту.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Вспышка коронавирусной инфекции COVID-19 [Електронний ресурс]. – Режим доступу: <https://www.who.int/ru>
2. Zoom: лидер в сфере конференц-решений согласно отчетам Gartner Magic Quadrant, 2019 г. [Електронний ресурс]. – Режим доступу: <https://zoom.us/>
3. Welcome to the Google Classroom Help Community[Електронний ресурс]. – Режим доступу: <https://support.google.com/edu/classroom/community/?hl=en&gpf=%23!forum%2Fgoogle-education>
4. Скайп помогает всегда быть на связи [Електронний ресурс]. – Режим доступу: <https://www.skype.com/ru/>
5. Делиться знаниями с миром ещё никогда не было так просто связи [Електронний ресурс]. – Режим доступу: <http://webinar.ua/>
6. Мы помогаем вам оставаться на связи [Електронний ресурс]. – Режим доступу: <https://clickmeeting.com/ru>.

РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗБОРУ КРИПТОВАЛЮТИ

Автоматизація розробки програм є важливим напрямком при підвищенні продуктивності праці. Вона дозволяє поліпшити якість продукції, оптимізувати процеси управління, а також відсторонити людину від виробництв, небезпечних для здоров'я. Автоматизація, за винятком найпростіших випадків, вимагає комплексного та системного підходу і особливих методів до вирішення різних завдань та проблем. Методи обчислень що застосовуються, іноді копіюють нервові і розумові функції людини. Тому весь цей комплекс засобів автоматизації зазвичай називають системами [1].

Важливим напрямком в області автоматизації програмування, крім мов програмування, бібліотек і пакетів прикладних програм, є створення інструментальних засобів розробки програмного забезпечення (ПЗ). При цьому сучасні засоби автоматизації процесу реалізації ПЗ дозволяють скоротити тривалість такого складного процесу, як ведення програмної документації. Цей процес реалізується з використанням генераторів документації програм. Вони аналізують мовні конструкції і коментарі у вихідному коді програм та їх конструкцій [1, 2].

Основний напрямок розвитку систем автоматизації це створення автоматичних систем, що здатні виконувати задані функції чи інші завдання без участі людини. Роль людини полягає у підготовці вихідних даних, виборі алгоритму (методу рішення) і аналізу отриманих результатів. В подібних системах передбачається поступово нарощувати захист від нестандартних подій (аварій) або способи їх обходу [2].

Для автоматизації процесу збору криптовалют розглянемо середовище розробки, мову програмування та систему вирішення Recaptcha v2, а також новітню її версію Recaptcha v3. Метою роботи є реалізація програмного забезпечення для збору криптовалют. Об'єктом дослідження – процес автоматизації збору криптовалют. Предметом – моделі, методи та технології автоматизації збору криптовалют.

Microsoft Visual Studio є серією продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інструментальних засобів. Вони дозволяють розробляти консольні програми і програми з графічним інтерфейсом та підтримкою технології

Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в стандартному, так і в керованому кодах для всіх платформ, що підтримуються [3].

Visual Studio побудована за архітектурою яка підтримує процеси використання доповнень (Add-Ins) чи плагінів від сторонніх розробників, що дозволяє розширювати можливості середовища розробки.

Вона включає редактор вихідного коду з підтримкою технології IntelliSense і можливістю використання найпростішого рефакторінга коду. Вбудований відладчик може працює як налаштувальник рівня вихідного коду або як налаштувальник машинного рівня. Решта інструментів, що вбудовуються, включають в себе редактор форм для спрощення реалізації графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнер схеми бази даних [3].

Visual Studio дозволяє створювати і підключати сторонні додатки з метою розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій та вихідного коду (Subversion і Visual SourceSafe), а також додавання нових наборів інструментів та редагування і візуального проектування коду на предметно-орієнтованих мовах програмування чи інструментів для інших подібних завдань циклу розробки програмного забезпечення (клієнт Team Explorer при роботі з Team Foundation Server) [4].

C# є об'єктно-орієнтованою мовою програмування з безпечною системою типізації для платформи .NET. Синтаксис C# близький до C++ і Java. Мова має строгую статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML [5].

Переїнявши багато від мов C++, Delphi, Модула і Smalltalk - C# виключає моделі, які є проблемними при розробці програмних систем, наприклад множинне спадкування класів. У C# існує єдина система типів які використовують загальний набір операцій і їх значення можна зберігати, передавати і обробляти схожим чином [6].

У третій версії Recaptcha вдосконалений поведінковий аналіз, то є та сама система advanced risk analysis, просунутого аналізу ризику. Тепер система працює «в тлі» і непомітно для користувачів.

Досить завантажити разом зі сторінкою бібліотеку Recaptcha і запустити greaptcha.execute в певний момент або відразу в момент завантаження сторінки [3].

З точки зору веб-майстра, головною відмінністю третьої версії є те, що за запитом через API сервер Recaptcha видає не бінарне значення, а оцінку в діапазоні від 0.0 (ймовірний бот) до 1.0 (ймовірна людина) для даного конкретного запиту. Відповідь надсилається у форматі JSON, приклад

наведено у лістингу 1.

Лістинг 1 – Приклад відповіді Recaptcha v3

```
{ "success": true|false, // whether this request was a valid reCAPTCHA token
  for your site
  "score": number // the score for this request (0.0 - 1.0)
  "action": string // the action name for this request (important to verify)
  "challenge_ts": timestamp, // timestamp of the challenge load (ISO format
  yyyy-MM-dd'T'HH:mm:ssZZ)
  "hostname": string, // the hostname of the site where the reCAPTCHA was
  solved
  "error-codes": [...] // optional }
```

Для відправки рекапчі на розпізнавання в CapMonster2, необхідно сформулювати запит, який включає url і sitekey сторінки, а також action. У прикладі в якості параметра action - 'homepage'. Після отримання токена необхідно передати його в функцію верифікації. Так як верифікація може статися в будь-який момент часу, потрібно перехопити запит на отримання токена і у відповіді підмінити його на отриманий токен з CapMonster.

Використання обраних методів реалізації програмного забезпечення включає мову програмування C# та середовище розробки Microsoft Visual Studio 2019, вона є крос-платформеною мовою з дуже зручним середовищем розробки та синтаксисом. Visual Studio дозволяє писати код своєю обраною мовою чи будь-якою з інших найбільш структурованих мов, використовуючи один і той же інтерфейс. Також швидке та якісне вирішення Recaptcha v3 є необхідним, тому обрано CapMonster та створено сніппет для успішної підмани такену при виконанні верифікації.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Вичугова А.А. Автоматизация процесса разработки программного обеспечения: методы и средства. / А.А. Вичугова // Прикладная информатика. – 2016. – Т.11., 3(63) – С.63-75.
2. Томашевський В.М. Моделювання систем. / В.М. Томашевський – Київ: BHV, 2005. – 349 с.
3. Тапскотт Д., Технология блокчейн / Д. Тапскотт, А. Тапскотт – 2016. – 448 с.
4. Fowler M. Patterns of Enterprise Application Architecture. / М. Fowler – Addison-Wesley, – 2012. – 544 p.
5. Характеристика інструментального ПЗ. Режим доступу: <https://lektsii.org/14-72572.html>.
6. Майлз Р. C# Programming Yellow Book, Cheese Edition. / Р. Майлз – Independently published, 2016. – 216 с.

УДК 004.652

Богатирьова Т.О.¹, Зеленцова І.Я.², Паромова Т.О.³

¹студ. гр. КНТ-517 НУ "Запорізька політехніка"

²канд. техн. наук, доц. НУ "Запорізька політехніка"

³старш. викл. НУ "Запорізька політехніка"

ОГЛЯД ЗАСОБІВ РОБОТИ З BIG DATA

У зв'язку з розвитком інформаційних технологій, світові обсяги накопиченої інформації значно зростають з кожним роком. Так, в 2018 році ця цифра становила цілих 33 зеттатерабайта і, за прогнозами, до 2025 року людство сформує 175 36 інформації, говориться в доповіді The Data Age 2025, яка була підготовлена аналітиками компанії IDC в листопаді 2018 року [1].

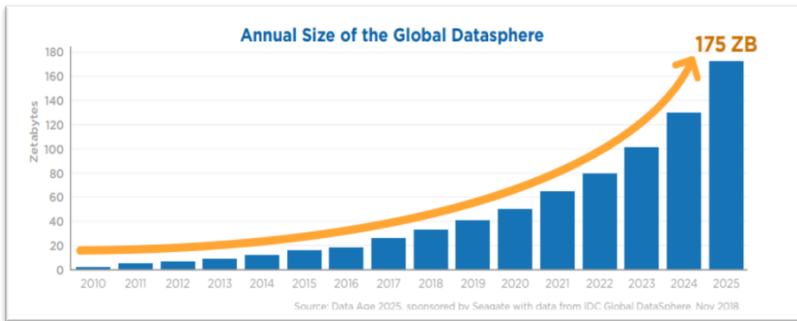


Рисунок 1 – Гістограма збільшення обсягів накопичених даних

Кількість користувачів глобальної мережі Інтернет значно зростає з кожним роком. Так, зокрема в звітах We Are Social і Hootsuite наводяться дані про глобальний стан цифрових технологій на 2019 рік: сьогодні в світі 5,11 мільярда унікальних мобільних користувачів, що на 100 мільйонів (2%) більше, ніж в попередньому році, в 2019 році аудиторія Інтернету налічує 4,39 мільярда чоловік, що на 366 мільйонів (9%) більше, ніж в січні 2018 року.

Це призводить практично до вибухового зростання обсягу накопиченої інформації, тому проблема обробки великих даних залишатиметься актуальною завжди і, як наслідок, призведе до значних якісних змін технологій обробки інформації. Як приклади джерел виникнення великих даних, зокрема наводяться дані, які безперервно надходять з вимірювальних пристроїв, події від радіочастотних ідентифікаторів, потоки повідомлень із

соціальних мереж, метеорологічні дані, дані дистанційного зондування Землі, потоки даних про місцезнаходження абонентів мереж стільникового зв'язку, пристроїв аудіо- і відеореєстрації. Очікується, що розвиток і початок широкого використання цих джерел ініціює проникнення технологій великих даних як в науково-дослідну діяльність, так і в комерційний сектор і сферу державного управління [2].

Big Data визначають як величезні обсяги неоднорідної цифрової інформації, що швидко надходить, які неможливо обробити традиційними інструментами [2]. Традиційні підходи до роботи з даними перестають бути ефективними в зв'язку з багатогранністю виникаючих завдань [3].

Визначальними характеристиками для великих даних, пов'язаних з особливостями їх обробки і аналізу, є принципи трьох, а потім розширених до п'яти і восьми V: volume, velocity, variety, veracity, viability, value, variability, visualization (фізичний обсяг, швидкість приросту даних і необхідності їх швидкої обробки, можливість одночасно обробляти дані різних типів, достовірність, життєздатність, цінність, мінливість, візуалізація відповідно).

На відміну від традиційних підходів, основні принципи роботи з великими даними такі:

- горизонтальна масштабованість;
- відмовостійкість;
- локальність даних.

Багато з того, що стосується великих даних, підходів, які використовуються для їх аналізу, насправді існує досить давно. Наприклад, обробка зображень з камер спостереження, коли аналізується не одна картинка, а потік даних.

Міжнародна консалтингова компанія McKinsey, що спеціалізується на вирішенні завдань, пов'язаних зі стратегічним управлінням, виділяє 11 методів і технік аналізу, які можна застосувати до великих даними:

- data Mining;
- краудсорсінг;
- змішання та інтеграція даних;
- машинне навчання;
- штучні нейронні мережі;
- розпізнавання образів;
- прогнозна аналітика;
- імітаційне моделювання;
- просторовий аналіз;
- статистичний аналіз;
- візуалізація аналітичних даних.

Необхідність порівняно недорогого зберігання і обробки гігантських

обсягів неструктурованої інформації привела до створення спеціалізованого програмного забезпечення, яке дозволило розподіляти дані по кластерам з сотень і тисяч вузлів, а також обробляти їх в паралельному режимі. Це призвело до створення Hadoop - відкритого фреймворка фірми Apache Software Foundation, який дозволяв створювати розподілені системи на базі відносно недорогого обладнання масового попиту. На його основі фірми - розробники Cloudera, MapR і Hortonworks пропонують свій дистрибутив. У 2011 році Hadoop помітили Oracle, IBM, Teradata і також включили в лінійки своїх програмних продуктів.

Для звичної роботи з інформацією за допомогою SQL на базі Hadoop розгортаються СУБД Hive або Impala. При потребі в можливостях NoSQL рішення - HBase.

Зберігати інформацію також можна в хмарі Amazon S3 або в NoSQL базі даних Cassandra, управляти ресурсами кластерів за допомогою Apache Mesos, видобувати і обробляти дані за допомогою Apache Spark, який здатний працювати і поза Hadoop. Spark останнім часом набирає популярність, так як обіцяє прискорення виконання розподілених програм обробки даних в порівнянні з Hadoop MapReduce до 100 разів. Може працювати як поверх HDFS під керуванням Hadoop YARN, так і поза фреймворка Hadoop на базі Cassandra, Amazon S3 і Google Cloud Storage (через Alluxio).

Дані, отримані з розподілених систем, обробляються в аналітичних інструментах SAS (Enterprise Miner), IBM (SPSS), Teradata (Aster Analytics) або Oracle (Advanced Analytics) або за допомогою маси інших комерційних та open-source рішень.

Big Data допомагають вирішувати різні завдання в різних сферах людського життя, серед них: рітейл, медицина, фінанси, промисловість, енергетика, туризм, екологія, розваги. Завдяки обробці та аналізу великого масиву даних, поліпшується якість товарів і послуг, з'являються нові можливості для розвитку бізнесу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. The Digitization of the World From Edge to Core [Електронний ресурс]. – Режим доступу: <https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf>
2. Big data [Електронний ресурс]. – Режим доступу: <https://intellect.icu/big-data-6821>
3. Big Data [Електронний ресурс]. – Режим доступу: <https://www.it.ua/ru/knowledge-base/technology-innovation/big-data-bolshie-dannye>

ПОРІВНЯЛЬНИЙ АНАЛІЗ ІГРОВИХ ДВИГУНІВ ДЛЯ НАСТІЛЬНИХ ПРИСТРОЇВ

За останні роки відеоігри зайняли значну частину ринку розваг та дозволя, посунувши кіно та музику. Великі компанії зайняли своє місце у ігровій індустрії. Вони спрямовані на створення відеоігор, які будуть чітко асоціюватися з певним жанром і приносити великий прибуток. Але невеликі розробники також мають шанси на створення прибуткової гри. Це стає можливим завдяки тому, що засоби розробки відеоігор стають дедалі доступнішими та досконалішими.

На відміну від великих компаній, які намагаються своїми іграми зайняти місце певного жанру, інді розробники намагаються створити незвичайну гру у незвичайній комбінації жанрів. Цей підхід є дуже ризикованим, тому не підходить для великих компаній.

Разом з розвитком комп'ютерних ігор розвивалися також засоби їх розробки. Одним із таких засобів є ігровий рушій, який значно полегшує процес розробки гри. Дуже важливим є етап вибору ігрового рушія, тому що невірно обраний рушій може ускладнити і затримати розробку, якщо рушій не призначений для жанру гри, що розробляється.

Вузько спеціалізовані ігрові рушії, які призначені для певних жанрів або, навіть, розроблюються під певну гру. Їх особливістю є повна оптимізація під певний жанр або гру, відсутність безкоштовної ліцензії та навчальних матеріалів у вільному доступі. Прикладами таких рушіїв є: Decima, X-Ray, Frostbite.

Універсальні ігрові рушії дозволяють створювати ігри різни жанрів для різних платформ. Вони, як правило, умовно безкоштовні та мають багато навчальних матеріалів. Cry Engine, Unity та Unreal Engine.

Існують ігрові рушії, які вже мають початкові ресурси, дії, та не вимагають знання мов програмування. Прикладами таких програм є GameMaker, Construct, RPG Maker.

Використання ігрового рушія є переважним вибором інді розробників. З переглянутих варіантів найбільш привабливими виглядають Unreal Engine та Unity, у зв'язку з великою кількістю навчальних матеріалів та низьким порогом входження. Проаналізувавши вся показники можна рекомендувати Unreal Engine як найбільш цілісний інструмент розробки гри.

УДК 004.42

Демченко М.М.¹, Точилін С.Д.²

¹студ. гр. КНТ-519м, НУ «Запорізька політехніка»

²канд. фіз.-мат. наук, доц. НУ «Запорізька політехніка»

КРОСПЛАТФОРМНИЙ ПРОГРАМНО-АПАРАТНИЙ КОМПЛЕКС ДЛЯ ЕЛЕКТРИЧНИХ ВИМІРІВ

Перспективним напрямком у плані доступності та досягнення високих експлуатаційних параметрів є створення засобів вимірів на базі персональних комп'ютерів [1]. При цьому розробка програмно-апаратних комплексів (ПАК) для електричних вимірів (ЕВ) є актуальною задачею.

На сучасному ринку електронних компонентів доступні мікроконтролерні плати Arduino. Їх можна використовувати як апаратну частину ПАК для ЕВ, що функціонує як система збору даних вимірів. Із цією метою плати програмують, зокрема, з допомогою скетчів Arduino IDE, які зберігаються у файлах формату *.ino.

У той час для створення кросплатформних програмно-апаратних комплексів з платами Arduino можна використовувати комп'ютерні додатки, які написані на мові програмування Processig.

У даній роботі був розроблений ПАК для ЕВ на основі плати Arduino UNO та програм Complex_DAS.ino і Complex_GUI.pde.

Програма Complex_DAS.ino дозволяла використовувати Arduino UNO як систему збору даних ЕВ.

Для забезпечення відображення результатів вимірів та керування роботою комплексу використовувався додаток Complex_GUI.pde, розроблений на Processing.

Основні характеристики програмно-апаратного комплексу:

1. Діапазон виміру напруги: 0-25 В.
2. Діапазон виміру струму: 0-3 А (для виміру струму використовувався датчик струму GY-MAX471).
3. Діапазон виміру опору: 0-10 Ком.

Надалі передбачається модернізувати розроблений програмно-апаратний комплекс, зокрема, розширити діапазони вимірів напруги та опору.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Богатырев С.Д. Генератор частоты на персональном компьютере / С.Д. Богатырев, Ю.А. Пильщикова, В.В. Родин // Фундаментальные исследования. – 2012. – № 11 (часть 4). – С. 909-912.

РОЗРОБКА СУМАТОРУ ЧИСЕЛ З РУХОМОЮ КРАПКОЮ НА FPGA

Широке використання дійсних чисел обумовлене тим, що при вирішенні значної частини задач фізики, математики або інформатики необхідно представляти результат з якомога більш високою точністю. Реалізація арифметики над числами з рухомою крапкою на програмованих логічних інтегральних схемах, зокрема на FPGA, має такі переваги, як гнучкість використання та низькі витрати на виробництво. Пристрій на FPGA можна пристосовувати для тих або інших потреб виробництва, що зменшує собівартість кінцевого продукту та надає можливість вирішувати задачі, для яких не існує готових рішень у вигляді представлених на ринку пристроїв.

Стандарт IEEE 754 визначає формат чисел з рухомою крапкою та методи арифметичних операцій з ними в обчислювальних машинах. Це найбільш поширений стандарт, який гарантує однаковість виконання операцій з числами з рухомою крапкою на програмному та апаратному рівнях. Двома основними типами чисел з рухомою крапкою в форматі IEEE 754 є тип float (одинарна точність) та тип double (подвійна точність).

В даній роботі розглядаються тільки нормалізовані числа з одинарною точністю, оскільки арифметичні операції над числами float та double не відрізняються. Авторами пропонується спосіб підвищення ефективності операції складання дійсних чисел з використанням конвеєризації [1]. Схема спроектована для реалізації на мікросхемах типу FPGA або ProASIC, що дозволяє використовувати архітектурні особливості даного базису, зокрема можливість паралельного виконання задач на одному кристалі. Базова структура суматора чисел з рухомою крапкою, розроблена відповідно до стандартного алгоритму обробки чисел в форматі IEEE 754. Структура орієнтована на проектування в базисі FPGA, тому має чітко виділені функціональні блоки, котрі можна поведінково визначити з використанням мови опису апаратури VHDL.

Для покращення характеристик суматора, його було конвеєризровано. Конвеєризована структурна схема є функціонально адекватною базовій структурі, але розділена на окремі рівні, які виконуються паралельно, а результат виконання операції кожного рівня не залежить від результату виконання операції попереднього рівня в один і той самий момент часу, це дозволяє виконувати операції складання/віднімання над декількома числами

одночасно. Що не тільки збільшує пропускну здатність, а й робить суматор придатним для використання в сучасних синхронних схемах. Кожний блок запропонованої конвеєрної структури на FPGA реалізовано, як окремий проект цифрового функціонального вузла, і таким чином, загальна задача розділена на окремі підзадачі, що полегшує експериментальне тестування та відлагодження складових усього пристрою.

Експериментальні дослідження виконані з використанням пакету EDA Quartus II [2] Розроблена схема була промодельована на FPGA сімейства Stratix III та на FPGA сімейства Cyclone III. Аналогом розробленої схеми є функціонально схожий пристрій фірми Altera. В порівнянні з пристроєм фірми Altera, що має такий самий функціонал, було збільшено тактову частоту на 37.6% на FPGA сімейства Stratix III (таблиця 1) та на 12.3% на FPGA сімейства Cyclone III.

Таблиця 1 – Порівняльна характеристика синтезованого конвеєрного пристрою на FPGA Stratix III та серійного пристрою фірми Altera

Пристрій	cycles	ALUTs	Dedicated registers	Fmax, MHz
ALTFP_ADD_SUB megafunction	7	592	376	218
Синтезований пристрій	7	541	373	300

Розроблений суматор має широку сферу використання, оскільки на даний час майже жоден з обчислювальних пристроїв не обходиться без обробки чисел з рухомою крапкою. На сучасному ринку представлений широкий вибір складних функціональних блоків, таких як модулі ІР (Intellectual Property). Але ж при їх використанні в проектах виникають певні проблеми, пов'язані як із ускладненням маршруту проектування, так і з підвищенням кінцевої вартості пристрою. Запропонована модель прискорювача є досить простою в сенсі реалізації на FPGA та може бути альтернативою використанню вбудованих помножувачів та процесорних ядер у випадках, коли складний функціонал зазначених пристроїв є надлишковим для конкретної задачі.

Обчислювальні функціональні вузли, реалізовані на FPGA, мають певні переваги порівняно з мікропроцесорами, в тому сенсі, що вони можуть бути спроектовані для конкретної задачі, що сприяє прискоренню обробки даних та зменшенню собівартості готового пристрою.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Танненбаум Э. Архитектура компьютера [Текст] / Э.Танненбаум – М.: Питер, 2009. – 844с.

2. Описание САПР Quartus II, Основные этапы проектирования СБИС ПЛ - Формирователь OFDM сигнала на плис стандарта 802.16d [Електронний ресурс]. – Режим доступу: <http://www.techexpose.ru/texob-544.html>

УДК 004.056.5

Дьячук Т.С.¹, Антонюк Д.С.²

¹ асист. НУ "Запорізька політехніка"

² студ. гр. КНТ-516 НУ "Запорізька політехніка"

ЗАХИСТ ANDROID-ЗАСТОСУНКУ ВІД ЗЛОМУ

При розробці мобільного застосунку слід враховувати, що дані, якими оперує цей застосунок, можуть представляти певний інтерес для третіх осіб. Міра цінності цих даних варіюється в широких межах, проте, навіть найбільш проста приватна інформація, наприклад, пароль входу у застосунок, вимагає опрацювання її захисту. Особливо це важливо у світі поширення мобільних застосунків на усі сфери електронних послуг, включаючи фінансові, банківські операції, зберігання і передачу особистих даних тощо. Задача, яка стоїть перед розробником застосунку – створити надійне програмне забезпечення, яке буде виконувати свої безпосередні функції, швидше та краще за інших і при цьому мати обов'язково захист від злому, крадіжки, копіювання [1].

Гаджет - невеликий пристрій, призначений для полегшення і удосконалення життя людини. Сьогоднішній ринок розумних гаджетів пропонує близько 80% смартфонів під керівництвом операційної системи (ОС) Android, що приводить до високого попиту на застосунки цієї ОС. Рівень піратства в екосистемі Android досить високий. Застосунок не тільки легко вкрасти - його легко зламати, відв'язати від сервісів перевірки, відключити рекламу (економічна вигода розробника) або навіть впровадити в нього «бекдор» (метод обходу стандартних процедур аутентифікації).

Існує достатньо уразливостей застосунків, наприклад: використання незахищених локальних сховищ; зберігання критично важливих даних в коді; застосування алгоритмів зі зберіганням приватного ключа; використання асиметричного алгоритму з приватним ключем, відомим серверу; використання самописних алгоритмів шифрування і захисту; передача критично важливих даних в зовнішнє середовище у відкритому виді; реверсивна інженерія алгоритмів, що представляють інтелектуальну цінність.

Розглянемо основні методи захисту застосунків: системи ліцензування, вбудовані в магазини застосунків, платформи залежні бібліотеки для перевірки ліцензії; обчислення на стороні сервера (Server-side computation);

запобігання декомпіляції (написання частин коду на «нативній» мові, обфускація коду [2], мінімізація коду); перенесення частини програми на сервер; шифрування передачі даних клієнт-сервер; зберігання даних у обробленому та/або шифрованому вигляді; мінімізація запитів до серверу; використання надійних хеш- та шифро- стандартів; заборона збереження даних на зовнішньому накопичувачі.

Не рекомендується вигадувати нові принципи шифрування або створення хеш функції, адже велика ймовірність прогледіти вразливість власного алгоритму і швидкість його підбору та злому. Тим не менш, користування декількома методами шифрування одночасно над одними і тими самими даними – не доцільне. Додаткові методи рішення та поради:

- найбільш вдалим варіантом призначеного для користувача коду є графічний код, цифровий (6 цифр або більше), - як додатковий варіант;

- HTTP-клієнт має бути налаштований на примусове використання захищеного каналу зв'язку (HTTPS);

- у релізній збірці застосунка мають бути відключені усі налагоджувальні функції, наприклад, опція `debuggable`, щоб уникнути можливості підключення до програми зовнішнім налагоджувальним застосуванням;

- на екранах застосунка, де розміщується приватна інформація користувача, буде не зайвою примусова заборона на програмне створення скриншотів вікна застосунка, а також відключення показу скриншотів в диспетчерові завдань;

- будь-яка приватна інформація може додатково перевірятися запитом особистого ключа користувача, заданого їм для входу в застосунок (якщо такий є);

- усередині коду застосунка рекомендується користуватися відносними шляхами (`getCanonicalPath`) замість абсолютних шляхів (`getAbsolutePath`);

- використовувані в застосунку відкриті компоненти (наприклад, `Service` або `Content Provider`) мають бути обов'язково закриті за допомогою прапора `exported = false` в маніфесті застосунка (`Android Manifest`). Це дозволить заборонити доступ до цих компонент з іншого застосунка.

Таким чином, в умовах, коли користувач або зловмисник може отримати повний доступ до застосунка, існує лише одне оптимальне рішення: слід зберігати як можна менше секретних даних на стороні клієнта, завантажуючи важливу інформацію з боку сервера, при необхідності. Використовувати захищений протокол. Забезпечити повний захист від злому неможливо, і з цим доводиться миритися. Однак система вважається надійною, якщо вартість даних виявляється нижче, ніж вартість їх отримання. Комбінація описаних вище способів захисту допоможе створити надійний та безпечний Android - застосунок.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Георгий Малюков. Безопасность данных в разработке мобильных приложений [Електронний ресурс] / George Malyukov // Habr. – Режим доступу: <https://habr.com/ru/post/327760> – Назва з екрана.

2. Денис Матвеев. Защита Андроид приложения от отладки [Електронний ресурс] / Denis Matveev // Cryptoworld. – Режим доступу: <https://cryptoworld.su/zashhita-android-prilozheniya-ot-otladki> – Назва з екрана.

УДК 004.056.53

Жигулін Д.І.¹, Головіна К.П.¹, Дьячук Т.С.²

¹ студ. гр. КНТ-526 НУ «Запорізька політехніка»

² асист. НУ «Запорізька політехніка»

ВИКОРИСТАННЯ БАГАТОФУНКЦІОНАЛЬНОГО МЕРЕЖЕВОГО СНІФФЕРА “INTERCEPTER-NG” У ВІДКРИТИХ WI-FI МЕРЕЖАХ

Метою роботи є аналіз прийомів сканування трафіку відкритих точок доступу Wi-Fi, а також пошук способів самостійного захисту від рук зловмисників, що можуть вкрасти особисті дані, документи та паролі.

Interceptor-NG – це програма для виконання атак "людина-посередині". Головною особливістю, яка виділяє Interceptor-NG серед подібних програм, є те, що спочатку вона була написана для Windows і прекрасно працює саме в цій операційній системі. Також до особливостей програми можна віднести графічний інтерфейс, в якому зібрані численні функції та опції, пов'язані з атакою людина-посередині [1].

У практичному сенсі, атака людина-посередині полягає у можливості переглядати передані іншими користувачами дані в локальній мережі. Серед цих даних можуть бути логіни і паролі від сайтів. Процес аналізу переданих даних називається сніффінгом. У процесі сніффінга Interceptor-NG вміє: перехоплювати логіни і паролі для входу на веб-сайти, відновлювати передані дані (файли), перехоплювати повідомлення деяких месенджерів, та показувати відвідані користувачем адреси. Робота з програмою “Interceptor – NG” не дає стовідсоткового результату, так як є нестабільною. Це означає, що здійснення атак може погіршити швидкість з'єднання інтернету та зв'язок сніффінг-програми з користувачем (можливо атаку доведеться повторювати декілька разів до «вдалого» результату).

Завдяки атаці, званої ARP спуфінгом, комп'ютер починає вважати шлюзом не роутер, а комп'ютер атакуючого. Атакуючий отримує запити від «жертви» і передає їх у пункт призначення (наприклад, запитує вміст веб-

сайту в Інтернеті), отримавши відповідь від пункту призначення, він направляє його «жертві». У цій ситуації атакуючий стає посередником – звідси інша назва атаки людина-посередині – «атака посередника». Для реалізації атаки ARP спуфінг необов'язково розуміти її деталі [2].

За допомогою функції “Smart scan”, ми можемо відстежити пристрої, які підключені до нашої точки доступу, обрати ціль для атаки та виконувати будь-які функції Interceptor-NG.

Функція «Traffic changer» — головною метою цієї функції є отримання переходів на сторінки або рекламу, які можуть містити в собі загрозливе програмне забезпечення. Використовується для підміни слова пошукового запиту користувача на будь-яке інше слово. Недоліком цієї атаки є те, що користувач одразу помітить, що його запит був підмінений. Важливо, щоб ключове слово та слово користувача містило однакову кількість символів.

Функція «Cookie killer» — це видалення cookie-файлів користувача для повторної авторизації з метою отримання логіна та пароля. Головною метою цієї атаки є вихід з аккаунту користувача на сторінку авторизації. Повторне введення даних авторизації призведе до отримання цих даних зловмисниками. Також працює і з SSL-з'єднаннями та за допомогою функцій мережесих атак “SSL MitM” та “SSL Strip”, видаляємо “cookie-файли” з браузера користувача.

Функція «FATE» – головною метою цієї атаки є підміна оригінальної сторінки сайту на фейкову, адреса якої буде відрізнятися мінімально. Усі дані які будуть введені користувачем будуть отримані зловмисниками. Таким чином, користувач вводить на підмінну сторінку авторизації свої справжні дані та бачить помилку авторизації. На другій спробі підмінна сторінка зникає, а ми отримуємо дані які були введені на підмінній сторінці. Важливим недоліком цієї атаки є те, що адреса підмінної сторінки має відрізнятися від оригіналу. Деякі користувачі можуть запідозрити, що їх трафік відстежують.

Всі наведені вище методи отримання чужої персональної інформації були проведені лише у дослідницьких та наукових цілях, задля з'ясування способів, що можуть завадити зловмисникам у їх справах та зможуть організувати безпечне користування Інтернетом. Ці способи не передбачають використання у реальному житті, тому що викрадення персональної інформації людини може привести до несення покарання людиною, що викрала ці дані.

Є декілька простих способів, які зроблять ваше підключення максимально безпечним: підключайтеся лише до відомих мереж, не вводьте додаткову інформацію, використовуйте VPN, використовуйте двухфакторну авторизацію, використовуйте надійні та безпечні браузери, такі як Google Chrome та Mozilla Firefox.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Interceptor-NG [Електронний ресурс]. – Режим доступу: <http://sniff.su/>
2. Багатофункційний сніфер [Електронний ресурс]. – Режим доступу: <https://interceptor-ng.com/>

УДК 658:004

Загоренко І.О.¹, Грушко С.С.²

¹ студ. гр. КНТ-537сп НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

СИСТЕМА ДИСТАНЦІЙНОГО КЕРУВАННЯ НА БАЗІ ARDUINO

Дистанційне керування - передача керуючого сигналу від оператора до об'єкта управління, що знаходиться на відстані, через неможливість передати сигнал безпосередньо, якщо об'єкт рухається, знаходиться на значній відстані або в агресивному середовищі і т.д.

Найширше застосування дистанційне керування знайшло в таких галузях:

- комп'ютерна техніка;
- охоронні системи і системи допуску;
- фото-, відео-, кінозйомка;
- військова справа;
- водний транспорт;
- промислове виробництво і будівництво;
- моделювання і т.д.

Для дистанційного керування електронними об'єктами використовується радіозв'язок, який можна організувати за допомогою технологій Wi-Fi, Bluetooth, інфрачервоного сигналу, або спеціальної радіоапаратури. Для дистанційного керування за допомогою Wi-Fi та Bluetooth, в якості передавача можна використовувати такі пристрої як смартфон або ноутбук, також приймач повинен мати спеціальний адаптер. При використанні інфрачервоного зв'язку необхідно щоб передавач та приймач могли фізично «бачити» один одного, що робить керування об'єктом за перешкодою неможливим. Радіоапаратура представляє собою два завершені пристрої: передавач, за допомогою якого здійснюється керування, та приймач, який приймає сигнали. Цей спосіб найкраще підходить для керування моделями техніки і може бути універсальним, тобто за допомогою однієї апаратури можна керувати будь-яким видом техніки (наземним, повітряним, водним), але така апаратура може бути досить дорогою.

Зазвичай для керування наземною або водною технікою достатньо лише

двох каналів (рух вперед, назад та повороти) і такий вид апаратури не дозволяє керувати додатковим обладнанням (звуком, світлом і т.д.). Для керування всім цим обладнанням за допомогою одного устаткування необхідно збільшувати число каналів, що приводить до подорожчання проекту. Одним зі способів вирішення цієї проблеми є використання Arduino.

Arduino - це платформа розробки електронних пристроїв, яка складається з самої плати і програмного забезпечення (Arduino IDE). До плати можна підключити велику кількість датчиків, лампочок, реле і т.д., і керувати ними вручну, за допомогою кнопок, потенціометрів або автоматично, в залежності від прошивки.

Для організації радіозв'язку на Arduino можна використовувати Bluetooth; для керування Arduino за допомогою смартфона або ноутбуку - Wi-Fi, що дозволяє керувати Arduino з будь-якого куточку планети, де є інтернет; використання спеціальних радіомодулів дозволяють забезпечити зв'язок між кількома Arduino. Для керування електричними двигунами для Arduino необхідний спеціальний драйвер двигунів, задачею якого є приймання сигналу від плати та передавання їх двигунам.

Для системи дистанційного керування об'єктом (моделлю корабля) пропонується використовувати наступні компоненти: для передавача – плату Arduino Nano, радіомодуль NRF24L01, два джойстики на основі потенціометрів; для приймача (власне моделі корабля) – плату Arduino Nano, радіомодуль NRF24L01, драйвер двигунів на основі мікросхеми L298N для підключення двох колекторних електродвигунів або одного крокового, електродвигун, сервопривід SG90 на 1,6 кг.

Використовуючи такий набір компонентів можна створити просту двоканальну апаратуру з пропорційним керуванням для управління наземною або водною технікою. Окрім цього використовуючи весь потенціал радіомодулів NRF24L01 та самої Arduino можна створити досить складну, багатоканальну (до 83 каналів) апаратуру для керування повітряною технікою, використовуючи потужні безколекторні двигуни та регулятори для них.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Белов А.В. ARDUINO: от азов программирования до создания практических устройств / А.В.Белов. – СПб.: Наука и техника, 2018. – 480 с.
2. Днищенко В.А. 500 схем для радиолюбителей. Дистанционное управление моделями / В.А. Днищенко. – СПб.:Наука и техника, 2007. – 464с.
3. Что такое Arduino: платформа для изучения электроники [Електронний ресурс] –Режим доступу:<https://amperka.ru/page/what-is-arduino>.
4. Радиомодуль NRF24L01: описание, подключение, схема [Електронний ресурс]. – Режим доступу: <https://3d-diy.ru/wiki/arduino-moduli/radio-modul-nrf24l01>.

ТЕХНОЛОГІЇ BIG DATA АНАЛІТИКИ В РОЗПОДІЛНИХ СИСТЕМАХ ОБЧИСЛЕНЬ

В сучасному світі дані постійно створюються споживачами Інтернет послуг, користувачами соціальних мереж, датчиками, реактивними двигунами, фондовими ринками та практично всім, що пов'язано з мережею. Ці дані зростають кількісно та змінюються у реальному часі. Аналіз даних також повинен здійснюватися в режимі реального часу під час збирання даних. Обсяг даних, які необхідно зберігати та аналізувати, продовжує розширюватися. Поява великих наборів даних вимагає застосування ефективних методів, технологій та інфраструктури для обробки даних та перетворення їх у діючу інформацію [1].

Основними поняттями Big Data аналізу є [2]:

- кластер – для пошуку схожих груп спостережень;
- асоціація – для пошуку спільного входження значень для різних змінних;
- регресія – для кількісної оцінки відношень між варіантами однієї чи кількох змінних.

Аналітика, проведена на великих наборах даних, може надати нові можливості і непередбачені тенденції, забезпечуючи краще представлення клієнтів і ринок у цілому. Точна аналітика клієнтів, виявлення випадків шахрайства та аналіз ризиків – все це вигоди від Big Data аналітики. Ці складні обчислення вимагають не лише великих запасів даних, але й малу затримку, високу пропускну спроможність обробки потоку.

На сьогодні основними інструментами та технологіями, які використовуються для Big Data аналітики є Python, R, Scala, Apache Spark, Hadoop, MapReduce, Cassandra, Kafka, алгоритми пошуку даних, комп'ютерне навчання, статистичні методи, NoSQL.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Олещенко Л.М. Технології Big Data / Л.М. Олещенко // Проблеми інформатизації та управління, 4(60), 2017 – С.57-64
2. David Haertzen. ETL Tools // The Analytical Puzzle: Profitable Data Warehousing, Business Intelligence and Analytics. – Technics Publications, 2012. – 346 p.

УДК 004.582

Іващенко Н. А.¹, Бредун А. Ю.², Дьячук Т. С.³

¹ студ. гр. КНТ-518 НУ «Запорізька політехніка»

² студ. гр. КНТ-528 НУ «Запорізька політехніка»

³ асист. НУ «Запорізька політехніка»

КОМПЛЕКС МОНІТОРИНГУ ПОКАЗНИКІВ ТЕМПЕРАТУРИ ТА ВОЛОГОСТІ СИСТЕМИ ОПАЛЕННЯ ТА КОНДИЦІОНУВАННЯ ПІДПРИЄМСТВА

Підприємство, основним напрямком роботи якого є прокладання комунікацій, використовує в своїй роботі обладнання, що потребує постійного контролю за показниками температури та вологості. Якщо значення температури в системі опалення не буде відповідати допустимому діапазону, то це призведе до вибуху, який стане причиною зупинки виробництва.

Існуючі на сьогоднішній день датчики, що працюють за такою технологією, не можуть правильно функціонувати з сучасними приладами, і тому їх неможливо використовувати на даному виробництві. Більшість таких датчиків продаються вже з заданими температурними значеннями та показником вологості, деякі з них можна синхронізувати з іншими пристроями. Також, необхідні нам системи встановлюють на етапі планування комунікацій опалення та кондиціонування, що в даний момент неможливо, і тому найкращим виходом для вирішення даної проблеми буде впровадження власного розробленого комплексу.

Для вирішення цієї проблеми авторами запропоновано пристрій, що порівнює температуру та вологість у виробничому приміщенні з вказаним користувачем діапазоном. Прилад відображає дійсні значення вимірюваних показників та відправляє їх в мобільний додаток, розроблений для цього пристрою. При несправній роботі системи опалення та кондиціонування буде спрацьовувати система оповіщення на пристрої та в мобільному додатку.

На рисунку 1 представлена структурна схема комплексу моніторингу показників температури та вологості системи опалення та кондиціонування підприємства, по якій ми визначаємо, що потрібно для датчика, як це буде працювати та як взаємодіяти. Система оповіщення спрацьовує, коли значення на датчиках наближається до граничних, що супроводжується повідомленням на телефон, та на самому пристрої загоряються діоди. Датчики температури та вологості вимірюють відповідні показники системи опалення та кондиціонування підприємства, та відображають їх на дисплей. За допомогою мобільного додатку користувач може слідкувати за роботою комплексу моніторингу.

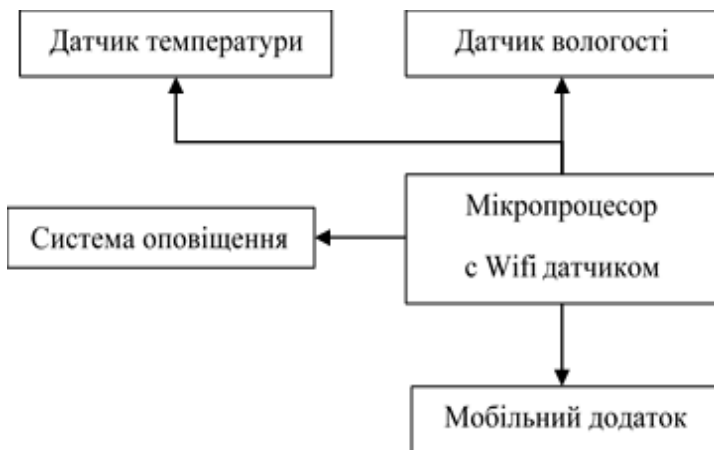


Рисунок 1 - Структурна схема комплексу моніторингу показників температури та вологості системи опалення та кондиціонування підприємства

Будь-яка обчислювальна машина, здатна автоматично виконувати деякі операції, має в своєму складі керуючий контролер - модуль, що описує логіку роботи пристрою. Контролер - це мозок машини. Звісно, чим складніше логіка роботи машини, тим "розумніше" повинен бути контролер [2].

Наш пристрій розроблено з використанням мікроконтролера Arduino з Wifi модулем та приєднаними до нього датчиками температури та вологості, а також відображенням виміряних значень на дисплеї [1]. Мобільний додаток був створений під мобільну платформу Android, за допомогою мови програмування Java.

Таким чином для вирішення поставленої задачі був спроектований комплекс для моніторингу показників системи, який значно полегшує роботу виробництва, дозволяючи постійно спостерігати за виміряними значеннями.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Петров І. В. Програмовані контролери. Стандартні мови та прийоми прикладного проектування / І. В. Петров. – М.: Солон-Пресс, 2004. – 255 с.
2. Аверченков О. Є. Схемотехніка: апаратура та програми / О. Є. Аверченков. – М.: ДМК Пресс, 2012. – 588 с.

УДК 004.584

Іващенко Н.А.¹, Федорова В.Ю.¹, Дьячук Т. С.²

¹ студ. гр. КНТ-518 НУ «Запорізька політехніка»

² асист. НУ «Запорізька політехніка»

BULLET JOURNAL НА ПЛАТФОРМІ ANDROID

Накопичення багатьох запланованих справ призводить до того, що про деякі з них людина забуває. Логічним рішенням цієї проблеми є запис планів. Для самоорганізації використовують блокнот або зошит. Цей спосіб дозволяє не тільки щось планувати, а і створювати різні списки так званих категорій, записувати свої думки та спогади. Існує така річ як bullet journal - це метод особистої організації, розроблений дизайнером Райдером Керроллом [1].

Для людей, які не бажають витратити час на самостійне оформлення своїх записів у такому форматі, існує компроміс – мобільні додатки. На сьогоднішній день смартфон використовується у більшості сфер життя сучасної людини, і тому зберігання всієї необхідної інформації в одному місці є найкращим варіантом.

Сучасний ринок мобільних додатків пропонує багато рішень на основі bullet journal. Але більшість з них має суттєвий недолік: дуже важко знайти те, що повністю підходить під потреби певної людини. Одні роблять акцент на організацію бюджету, другі - на формування здорових звичок, треті - на планер справ.

Для того щоб втілити ідею ідеального додатку для реалізації методу bullet journal ми пропонуємо власний додаток на платформі Android. Для реалізації проекту потрібно володіти мовою програмування Java та вміти працювати з базами даних. Розробка програм під Android проводиться всередині Android Studio, вона оснащена візуальним (графічним) та текстовим редактором [2].

На рисунку 1 представлена структурна схема Bullet Journal на платформі Android, яка зображає основний функціонал додатка. Через базу даних зорганізується вся робота програми. Інші блоки не пов'язані між собою, а дані для них беруться тільки з бази даних. Блок нотаток має функцію, яка дозволяє призначати категорії для кожного запису, що дає можливість для швидкого пошуку. Розділ планувальник - для створення подій, які прив'язані до певного часу або місця. Блок здоров'я надає користувачеві можливість відстежувати кількість випитої води та пройдених кроків. Блок фінансів дозволяє вести облік доходів / витрат. Блок авторизація створений для можливості збереження даних користувача.



Рисунок 1 - Структурна схема Bullet Journal на платформі Android

Для реалізації потрібного функціонала розробники створюють платні версії додатку, котрі не всі можуть собі дозволити, в той час як наша програма буде абсолютно безкоштовною. А також дозволить поєднувати в собі найбільш розповсюджені функції, які частіше всього знаходяться в окремих додатках, що суттєво ускладнює процес створення необхідних записів. Для вирішення поставленої задачі ми розробляємо додаток, що повністю буде задовольняти функціонал, який нам потрібен.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Райдер К. Bullet Journal метод. Переосмислити минуле, упорядкуй сьогодні, проєктуй майбутнє / К. Райдер. – Київ:Форс, 2019. – 336 с.
2. Android Firebase [Електронний ресурс]. – Режим доступу: <https://itproger.com/course/android-firebase>

СИСТЕМА КОМПРЕСІЇ ДАНИХ ДЛЯ WEB

Проблема передачі великих обсягів даних є актуальною в умовах інтенсивного розвитку Інтернету (зокрема соціальних мереж та сценарно-складних сайтів, кількість яких неухильно зростає). Таким чином, дослідження в області компресії даних різних типів, є актуальним завданням, вирішення якого дозволяє підвищити ефективність використання пропускну здатності мереж без істотних матеріальних витрат.

Завдання дослідження:

1. Дослідити існуючі програми компресії даних, аналізуючи їх структуру, ефективність і швидкість стиснення, вибрати оптимальний метод компресії високонадлишкових даних і графічних файлів, розробити систему стиснення даних для Web на основі власної варіації обраного методу компресії та порівняти її з існуючими аналогами. Вивчити найближчі аналоги системи стиснення даних (архіватори і трафік-компресори).

2. Проаналізувати існуючі універсальні алгоритми компресії без втрат і методи контекстного моделювання, а також їх основні характеристики, що впливають на швидкість і ефективність стиснення.

Результати дослідження:

1. У процесі роботи над проектом були проаналізовані існуючі універсальні алгоритми стиснення без втрат (різні варіанти арифметичного кодування такі як інтервальне кодування і адаптивне арифметичне кодування) і методи контекстного моделювання, а також їх основні характеристики, що впливають на швидкість і ефективність стиснення, зокрема проведено аналіз методів оцінки ймовірностей утечі та способів підвищення точності оцінок в контекстних моделях високих порядків. На підставі проведених досліджень було з'ясовано, що алгоритми передбачення за частковим збігом (окремий випадок контекстного моделювання) є оптимальною по співвідношенню ефективність/швидкість компресії основою для універсальних систем компресії даних.

2. Дослідження виявили, що оптимальними порядками моделі для текстів на природних мовах є порядки 4, 5 і 6. Тому для високонадлишкових даних була створена власна варіація алгоритму передбачення за частковим збігом.

3. Для малонадлишкових даних (графічних файлів) на основі досліджень їх внутрішньої структури був розроблений власний алгоритм

стиснення без втрат, що ґрунтується на таких базових методах стиснення, як символвне подавлення, заміна і відносне кодування.

Висновки:

1. При використанні системи компресії даних, обсяг переданих по мережі даних скорочується в середньому на 80% у разі текстової інформації, на 70% для виконуваних файлів, на більш ніж 90% для баз MS Access і на 60% в разі невеликих зображень, і все це при прийнятній швидкості компресії/декомпресії.

2. Система компресії даних не має жорстких вимог до програмного і апаратного забезпечення, займає невеликий об'єм і проста у використанні. Програма може бути модифікована і вдосконалена шляхом додавання нових методів компресії та оптимізації існуючих.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Ватолин Д., Ратушняк А., Смирнов М., Юкин В. – М.: ДИАЛОГ-МИФИ, 2003. – 384 с.

2. Вернер М. Основы кодирования. Учебник для ВУЗов. / М. Вернер – М.: Техносфера, 2004. – 288 с.

3. Сэломон Д. Сжатие данных, изображений и звука / Д. Сэломон – М.: Техносфера, 2004. – 368 с.

4. Пахомов С. Сравнение 64-битных архиваторов / С. Пахомов // Компьютер Пресс, – 2013 – №3 (279) – С. 52–64.

5. Кириченко В.В. Аналитический обзор алгоритмов сжатия цифровой информации / В.В. Кириченко // ПФМТ – 2016 – №2 (27) – С. 77–83.

УДК 004.31

Котенко А.О.¹, Зеленьова І. Я.², Голуб Т.В.³

¹ студ. гр. КНТ-619м НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

³ асист. НУ «Запорізька політехніка»

РОЗРОБКА ДРУКОВАНОЇ ПЛАТИ БЛОКУ УПРАВЛІННЯ В АВТОМАТИЗОВАНІЙ СИСТЕМІ КОНТРОЛЮ РІДИНИ

Друковані плати є основним елементом електронної апаратури, виконуючи функції несучої конструкції і комутаційного пристрою на різних рівнях розукрупнення апаратури:

- в мікрозбірках;
- в осередках;

- в комутаційних (монтажних) панелях.

Друковані плати широко застосовуються в побутовій техніці, апаратурі засобів зв'язку, обчислювальній техніці, в системах автоматизації, контрольно-вимірjuвальної апаратури, в медичному приладобудуванні, в автомобільній промисловості, в інших областях промислової електроніки, в авіаційній, космічній промисловості, в спецтехніці, в міському комунальному господарстві (для засобів контролю витрати води, газу, електрики, палива та ін., екологічного контролю води, повітря, землі по радіаційним, фізичним, механічним і хімічно параметрам) [1].

Мета проекту – створення друкованої плати блоку управління насосом для автоматизованого і ручного контролю за рідиною.

Блок схема проекту показана на рисунку 1.

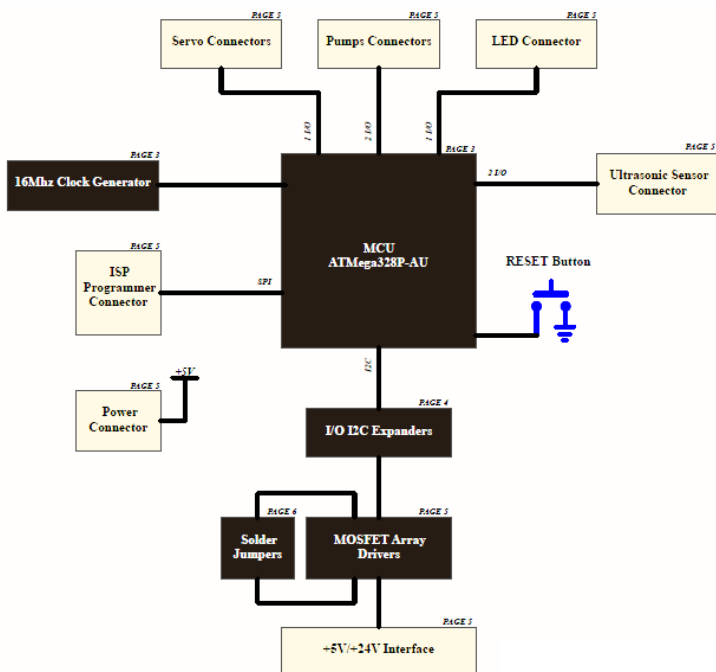


Рисунок 1 – Структурна схема блоку управління насосом

Основним керуючим елементом є мікроконтролер Atmega328, тактування якого виконується за допомогою кварцового резонатора на 16МГц. Перезавантаження виконується за допомогою кнопки Reset.

Для з'єднання блоку управління насосом з керуючим блоком

використовується паралельний інтерфейс (+5V/+24V Interface). Друкована плата була розроблена для двох варіантів логічних рівнів інтерфейсу – 5В та 24В. Для сумісності з 24В логічними рівнями була використана мікросхема збірки на польових транзисторах (MOSFET Array Drivers). Під мікросхемою розташовані паяльні перемички для сумісності з 5В версією інтерфейсу (Solder Jumps), які з'єднують з входами і виходами збірки.

Для мінімізації використаних входів/виходів мікроконтролера була використана мікросхема для їх розширення, яка контролюється мікроконтролером за допомогою інтерфейсу I2C (I/O I2C Expander).

На друкованій платі розташовані такі з'єднувачі:

- Power Connector – з'єднувач для підключення живлення;
- ISP Programmer Connector – з'єднувач для підключення ISP програматора;
- Servo Connector – з'єднувач для підключення сервоприводу (для ручного контролю);
- Pumps Connectors – з'єднувач для підключення насосу (для автоматизованого контролю);
- Ultrasonic Sensor Connector – з'єднувач для підключення ультразвукового датчика (для автоматизованого контролю);
- LED Connector – з'єднувач для підключення світлодіоду.

На рисунку 2 представлений верхній шар друкованої плати (Top Layer)

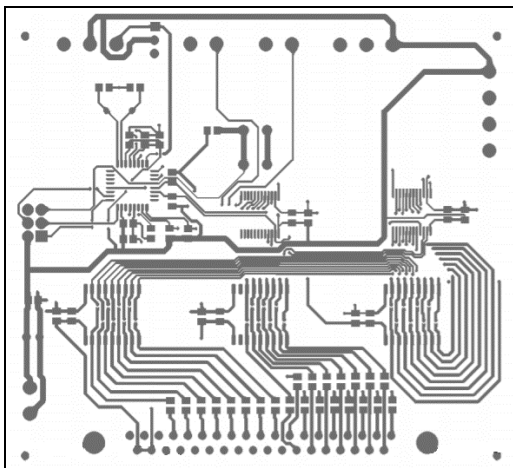


Рисунок 2 - Верхній шар друкованої плати

У ході виконання проекту була розроблена друкована плата блоку

управління насосом, що має ручне та автоматизоване керування й призначена для контролю за рівнем рідини. Друкована плата була виготовлена, спаяна та протестована. На даний момент плата використовується у системі яка розташована в університеті Ільменау в лабораторії GOLDi Labs. Плата використовується у навчанні студентів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Пирогова Е.В. Проектирование и технология печатных плат / Е.В. Пирогова – М.: ФОРУМ: ИНФА-М, 2005. – 560 с.

УДК 004.8

Куликовська Н.А.¹, Азаров В.В.²

¹ асист. НУ «Запорізька політехніка»

² студ. гр. КНТ-537сп НУ «Запорізька політехніка»

ПОШУК ЗАПОЗИЧЕНОЇ ІНФОРМАЦІЇ В ІНТЕРНЕТІ

Теперішній розвиток інформаційних технологій дав користувачам Інтернету доступ до великої кількості інформації. Це призвело до зростання її дублювання. Особливо це помітно в таких галузях, як освіта та ЗМІ. Таким чином, задача знаходження подібних запозичень набуває значущу актуальність. Документи з запозиченою інформацією поділяються на «чіткі» та «нечіткі» дублікати. Чіткий дублікат – точна копія оригінального документа. Нечіткий дублікат – документ, в якому частково змінено зміст та/або форматування. [1] Задача пошуку чітких дублікатів є тривіальною на відміну від пошуку нечітких.

Усі методи пошуку нечітких дублікатів можна умовно поділити на:

1. Синтаксичні методи (в яких здійснюється вибір послідовностей символів, слів та/або речень).

2. Лексичні (семантичні) методи (в яких здійснюється вибір представницьких мовних одиниць).

Основним синтаксичним методом є метод шинглування, коли вхідний документ представляють набором усіх підланцюгів послідовностей слів визначеної довжини. Такі ланцюги слів, що вибираються з певним зміщенням по лінійній структурі тексту, називають шинглами. Кількість можливих побудованих шинглів для одного речення дорівнює: кількість_слів_речення – довжина_шинглу + 1. Кожному ланцюгу співставляється хеш-код, при виборі якого забезпечуються наступні важливі властивості: рівність ланцюгів гарантує рівність кодів (тобто кодування є хеш-функція), а рівність кодів каже про високу подібність ланцюгів. Найбільш поширеними є хеш-коди

SHA1 та Rabin.

В методах лексичного типу реалізується відбір множини представницьких слів, виходячи з показників значимості цих слів. У такій множині не включаються слова зі заздалегідь фіксованого списку стоп-слов. Існує ряд методів, які засновані на лексичному представленні тексту: метод I-Match (обчислюється сигнатура (хеш-значення) для слів зі середнім значенням інверсної частоти слів в документах IDF), метод «опорних» слів (за визначеними правилами для кожного документа формується множина слів, для яких обчислюється сигнатура).

Для знаходження запозиченої інформації також використовують алгоритми, що побудовані на принципах інформаційного пошуку: TF, TF*IDF, TF*RIDF, OptFreq.

УДК 004.8

Куликовська Н.А.¹, Азаров В.В.²

¹ асист. каф. КСМ НУ «Запорізька політехніка»

² студ. гр. КНТ-537сп НУ «Запорізька політехніка»

АНАЛІЗ ЯКОСТІ ТЕКСТУ

У мережі Інтернет існує безліч сайтів, наповнених статтями різних якості та спрямованості: одні написані з метою надання користувачам інформації, інші - з метою продати той чи інший товар. Для оцінки подібних статей, необхідно визначити критерії: в рамках поставленого завдання їх зручно представити у вигляді чисел: від 0 до 1, де 0 - найгірший результат за конкретним критерієм. SEO-фахівці, що займаються наповненням сайтів контентом, нерідко перевіряють написані статті за допомогою різних інтернет-сервісів.

Сервіс Google Search Console дозволяє проаналізувати текст на водність, перевірити його орфографію і показати самі часто вживані словосполучення в тексті. Даний сервіс зручний для SEO фахівців, так як він дозволяє зробити перевірку написаних статей простіше. Тут, як і в інших системах, застосовується наступний алгоритм обчислення водності: обчислюється число стоп-слів в тексті, після чого отримане значення ділиться на загальне число слів. Схожа система SEO-CRM., реалізує такий же алгоритм для аналізу тексту, але з використанням максимального і мінімального меж значення оцінки: в даному веб-додатку тексти, більш ніж на 60% складаються з стоп-слів, отримують максимально негативну оцінку, так як вони не несуть в собі ніякої інформації.

В таблиці 1 наведені основні параметри якісного тексту з діапазоном

норми.

Таблиця 1 - Параметри якісного тексту

№	Показник	Діапазон норми, %	На що вказує
1	Унікальність	90-100	Відсутність/наявність копій тексту
2	Нудота	9	Співвідношення повторюваних слів до загального обсягу тексту
3	Водність	40-60	Наявність слів, що не несуть корисної інформації
4	Заспамленість	3-5	Частота вживання ключових слів

УДК 004.353

Лубянов Д. В.¹, Дьячук Т. С.²

¹ студ. гр. КНТ-526 НУ "Запорізька політехніка"

² асист. НУ "Запорізька політехніка"

ПРОГРАМУВАННЯ ПРИСТРОЇВ ДИСТАНЦІЙНОГО КЕРУВАННЯ НА ARDUINO

Актуальною залишається проблема дистанційного керування побутовими пристроями за допомогою фірмових пультів, які часто губляться, займають багато місця та періодично розряджаються. Оперувати великою кількістю пультів незручно, а інколи неможливо. Людям з обмеженими можливостями та людям похилого віку дуже складно опанувати новий пристрій. Для рішення даної проблеми необхідно замінити усі пульти дистанційного керування універсальним пультом або смартфоном.

Для пультів з декількома функціями виконується частотна модуляція несучого сигналу та кодування переданих команд, для цього використовується цифрова обробка. Мікросхема передавача (в пульті) модулює і кодує сигнал, що передається. А у приймачі проходить його демодуляція та декодування. Після демодуляції отриманого сигналу застосовуються відповідні частотні фільтри для поділу сигналів.

Для програмування пристроїв дистанційного керування необхідне інтегроване середовище розробки (IDE) та інтегральна схема для налаштування. В ході аналізу, була обрана програмована логічна інтегральна

схема (ПЛІС) Arduino UNO та IDE Arduino для програмування [1]. Програми пишуться на мові C/C++, компілюються та завантажуються до ПЛІС. Даний варіант відрізняється від конкурентів простотою налаштування, зручною ріп-компоновкою та відносною дешевизною. Крім цього, для отримання та передачі сигналу необхідні додаткові модулі: ІЧ-приймач VS1838B, ІЧ-світлодіод, Bluetooth-модуль HC-05 (підключення модулів до плати здійснюється за схемою розпіновки).

Спочатку необхідно зчитати код від пультів. До IDE завантажується відповідна програма (скетч) із бібліотеки IRremote [2]. Після компіляції відкривається моніторинг порту. Користувач по черзі натискає необхідні кнопки на пульті, відбувається зчитування коду за допомогою ІЧ-приймача та моніторинг порту відображає їх код на екрані (рис. 1).

Далі відбувається програмування УПДК за допомогою ПЛІС Arduino UNO. Програмованим пристроєм був обраний універсальний пульт дистанційного керування (УПДК) RM-L810 Universal. Він відрізняється багатим функціоналом, який повністю задовольняє наші потреби та ергономічним дизайном. У верхній частині присутні 10 кнопок перемикання між різною побутовою технікою. Програмування доступне завдяки функції Learn Function та здійснюється крок за кроком по інструкції (пульт містить як ІЧ-передавач, так ІЧ-приймач). Після написання коду програми, підключається бібліотека IRremote та записаний сигнал по чергово передається з ІЧ-передавача ПЛІС на ІЧ-приймач пульта в режимі Learn Function. Користувач сам визначає на яку кнопку записати отриманий сигнал.

```
Decoded NEC: 9966708F (32 bits)
Raw (68): 8950 -4400 600 -1600 650 -450 650 -500 600 -1600 650 -1600 600 -500 600 -450 7
9966D02F
Decoded NEC: 9966D02F (32 bits)
Raw (68): 8950 -4350 600 -1650 600 -500 600 -500 600 -1600 650 -1600 600 -500 650 -450 6
FE50AF
Decoded NEC: FE50AF (32 bits)
Raw (68): 8950 -4450 600 -500 600 -550 600 -500 600 -500 650 -450 650 -500 600 -550 600
FED02F
Decoded NEC: FED02F (32 bits)
Raw (68): 9000 -4400 600 -500 650 -500 600 -500 650 -450 650 -500 650 -500 600 -500 600
FE2AD5
Decoded NEC: FE2AD5 (32 bits)
Raw (68): 9000 -4400 600 -500 650 -500 600 -500 650 -500 600 -500 650 -500 600 -500 600
FFFF15
```

Рисунок 1 – NEC та Raw коди отриманих сигналів

Для керування пристроями за допомогою смартфона (з обов'язковою технологією ІЧ-сигналу) необхідне налаштування та програмування Android-додатку. Після завантаження додатку обираємо нашу плату у списку доступних пристроїв для підключення. При чому відпадає необхідність

ручного програмування кожної кнопки. Користувач сам визначає код необхідної кнопки в лістингу IDE Android-додатку. Принцип роботи: сигнал транслюється на Bluetooth-модуль плати, відпрацьовує завантажений скетч, зчитується отриманий код, змінюється на необхідний та через ІЧ-передавач сигнал передається до ІЧ-приймача пристрою [3].

Для тестування було обрано шість пристроїв: два різних телевізори, ТВ-приставка, CD-плеєр, DVD-плеєр та світлодіодна стрічка. Сигнали подавались коректно, тому пристрої відпрацьовували правильно. Програмування та налаштування займає відносно небагато часу, а всі комплектуючі обійдуться не дорожче за фірмовий пульт від виробника побутової техніки. Керування побутовою технікою доступне трьома способами – за допомогою будь-якого пульта, універсального пульта та смартфона.

Перший спосіб відрізняється дешевизною використання (для цього варіанту підходить будь-який непотрібний пульт). Сигнал з такого пульта відправляється на ІЧ-приймач ПЛІС, скетч ретранслює отриманий сигнал на необхідний та відправляє на ІЧ-приймач керованого пристрою. Підходить як тимчасова заміна фірмового пульта.

Другий спосіб здійснюється без зайвої ланки ПЛІС Arduino UNO. Всі необхідні сигнали пультів збережені в ПЗУ УПІДК. Програмування здійснюється одноразово і не потребує додаткових маніпуляцій.

Третій спосіб відрізняється складністю програмування, однак не потребує додаткових витрат. Керування здійснюється за допомогою смартфона з опціональною підтримкою технології ІЧ-сигналу. Крім написання скетчу в IDE Arduino необхідно написати лістинг програми в IDE Android Studio, IntelliJ IDEA тощо. Спосіб для досвідчених користувачів.

Висновки. В результаті користування даними пристроями дистанційного керування був відчутний комфорт та легкість управління побутовою технікою. Комплекс пристроїв стане у пригоді не лише звичайним користувачам, але і людям з обмеженими можливостями або людям похилого віку, які неспроможні використовувати декілька приладів для керування побутовою технікою за допомогою фірмових пультів або мають складнощі зі звиканням до нового пристрою керування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Плати Arduino UNO [Електронний ресурс]. – Режим доступу: <https://doc.arduino.ua/ru/hardware/Uno>
2. Уроки Arduino для начинающих и профессионалов [Електронний ресурс]. – Режим доступу: <https://arduinoplus.ru/lessons/>
3. Arduino и Android [Амперка / Вики] [Електронний ресурс]. – Режим доступу: wiki.amperka.ru/android-и-bluetooth

ВИКОРИСТАННЯ СИСТЕМИ JIRA ДЛЯ ТЕСТУВАННЯ ТА КЕРУВАННЯ ПРОЕКТАМИ

Розробка програмного забезпечення являє собою складний процес, в якому беруть участь багато виконавців. Цей процес можна розбити на цілий ряд завдань, від вирішення яких залежить успіх розробки в цілому. На цьому шляху може виникнути безліч помилок, які допускають розробники програмного забезпечення. Відстеження і виправлення цих помилок в ручному режимі є трудомістке завдання, яке потребує значного часу, що робить увесь процес неефективним. Автоматизувати процес виправлення помилок допомагають системи відслідковування помилок. Ці системи є (англ. Bug tracking system - BTS) прикладними програмами, які розроблені з метою допомогти розробникам програмного забезпечення (програмістам, тестувальникам і ін.) враховувати і контролювати помилки (баги), знайдені в програмах, а також стежити за процесом усунення цих помилок [1]. При фіксації помилки необхідно зібрати про неї максимальну кількість доступної інформації. Необхідно бути точним у спостереженнях. Особливо це стосується звітів про помилки, що приходять від користувачів. Як правило, BTS дозволяє зберігати інформацію про помилку в наступному вигляді: хто повідомив про проблему; дата і час, коли була виявлена проблема; серйозність проблеми; опис неправильної поведінки програми; хто займається вирішенням цієї проблеми; стан помилки. На практиці використовується велика кількість різних систем стеження за помилками, серед яких є вільно поширювані і платні.

Серед вільно розповсюджуваних можна відзначити Bugzilla (Багзілла) - вільна система відслідковування помилок (багтрекінга) з веб-інтерфейсом. Система Redmine - безкоштовна і має відкритий вихідний код веб-додаток, що використовується для управління версіями і пошуку помилок. Вона дозволяє користувачам управляти безліччю проектів і підпроектів. Система GNATS - вільна система відслідковування помилок включає в себе набір інструментів для відстеження програмних помилок (багів), про які повідомляють користувачі на центральному сайті. Вона вирішує проблему управління повідомленнями і взаємодії з користувачами різними способами. GNATS зберігає всю інформацію про повідомленій проблемі в її базах даних і надає інструменти для запитів, редагування і супроводу баз даних. Серед

платних систем слід відзначити системи YouTrack, Bontq, Jira.

Розглянемо найбільш затребувану серед платних систем стеження за помилками розробку компанії Atlassian Jira.

Jira - комерційна система відслідковування помилок, призначена для організації взаємодії з користувачами, хоча може використовуватись і для управління проектами. Jira є одним з двох основних продуктів компанії Atlassian (поряд з вікі-системою Confluence). Має веб-інтерфейс. Система базується на Java EE і працює на кількох популярних системах управління базами даних і операційних системах. Основний елемент обліку в системі - завдання (англ. Ticket або issue). Завдання містить назву проекту, тему, тип, пріоритет, компоненти і зміст. Завдання може бути розширена додатковими полями (також і нові призначені для користувача поля можуть бути визначені), додатками (наприклад - фотографіями, скріншотами) або коментарями. Завдання може редагуватися або просто змінювати статус, наприклад, з «відкритий» в «закритий». Які переходи між станами можливі, визначається через настроюється потік операцій. Будь-які зміни в задачі записуються в журнал.

Jira має велику кількість можливостей конфігурації: для кожної програми може бути визначений окремий тип завдання з власним workflow, набором статусів, одним або декількома видами уявлення (англ. Screens). Крім того, за допомогою так званих «схем» можна визначити для кожного індивідуального Jira-проекту власні права доступу, поведінку і видимість полів і багато іншого. Завдяки універсальному підходу можна застосувати Jira для багатьох непрофільних завдань, наприклад, управління вимогами, управління ризиками, аж до реалізації невеликої системи бронювання, автоматизації процесу рекрутингу. Для інтеграції з зовнішніми системами підтримуються інтерфейси SOAP, XML-RPC і REST. Поставляється із засобами інтеграції з такими системами управління версіями як Subversion, CVS, Git, Clearcase, Team Foundation Server, Mercurial і Perforce.

Існують доповнення, що дозволяють вбудувати Jira в інтегровані середовища розробки, у тому числі Eclipse і IntelliJ IDEA. Розробники розширень можуть викладати плагіни для продажу на спеціальний розділ сайту Atlassian.

Для реалізації всіх можливостей JIRA слід використовувати її доповнення. На сьогоднішній день Atlassian пропонує понад 2600 доповнень. Наприклад: Zephyr для JIRA дозволяє користувачам організовувати, керувати, шукати, створювати, планувати, виконувати і повідомляти про дії з тестування проектів, які активно управляються в JIRA. Але Zephyr має обмеження:

- неможливо імпортувати результат тесту,
- неможливо експортувати всі тест-кейси та результати їх виконання в

один файл;

- складно імпортувати існуючі тестові приклади з Excel, і для цього потрібно використовувати деякі зовнішні настройки;

- звітність може бути виконана тільки для одного конкретного циклу ітерації;

- накопичувальний звіт не може бути створений для всіх циклів тестування ітерації;

- тестові приклади мають лише 3 стандартних стовпця, неможливо налаштувати формат тестового сценарію;

- інтеграція пакетів автоматизації може бути виконана за допомогою Selenium RC;

- інтеграція з Selenium Web Driver не реалізована.

Плагін JIRA Toolkit - це унікальний набір розширень JIRA. Він містить велику колекцію корисних полів, що настроюються, допомагає фільтрувати інформацію, яка в іншому випадку була б недоступна, надає простий спосіб обходу обмежень JIRA або рішень деяких конкретних проблем.

JIRA Misc Workflow Extensions використовується для реалізації складних процесів в JIRA Server і JIRA Cloud.

JIRA Charting надає різні графіки і звіти для JIRA, тобто можливість для візуального подання проекту в різних контекстах.

Широке розповсюдження системи JIRA можна пояснити її перевагами: система має функції, які недоступні в будь-якому іншому інструменті; користувачі отримують останні оновлення про хід виконання проектів; система просто налаштовується та розширяється; інтерфейси JIRA прості і зручні; розробка виконана на основі сучасних веб-технологій; легко інтегрується з найпопулярнішими інструментами.

Таким чином можна констатувати, що JIRA має велике розмаїття доповнень, що робить її найпотужнішим інструментом для розробки, управління проектами та управління тестуванням. Багато відомих компаній і установ використовують JIRA (Nokia, LG, Європейська комісія, Європейський парламент, Гарвардський і Стенфордський університети і т.д.). Але слід відмітити і недоліки системи: важкість налагодження; складність засвоєння ефективного використання JIRA; мобільний додаток JIRA має обмежену функціональність; занадто складна для малого бізнесу і невеликих команд.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Jira (программное обеспечение) [Электронный ресурс]. – Режим доступа: [https://ru.bmstu.wiki/Jira_\(программное_обеспечение\)](https://ru.bmstu.wiki/Jira_(программное_обеспечение))

2. Лучший инструмент разработки для agile-команд [Электронный ресурс]. – Режим доступа: <https://www.atlassian.com/ru/software/jira>

3. Сравнение систем отслеживания ошибок [Электронный ресурс]. –

Режим доступа: <https://ru.wikipedia.org/wiki/>

4. Для чего используется Jira? [Электронный ресурс]. – Режим доступа: <https://www.atlassian.com/ru/software/jira/guides/use-cases/what-is-jira-used-for>

УДК 004.384

Луценко Н.В.¹, Паромова Т.О.¹, Івашенко Н.А.², Бредун А.Ю.³

¹ старш. викл. НУ "Запорізька політехніка"

² студ. гр. КНТ-518 НУ "Запорізька політехніка"

³ студ. гр. КНТ-528 НУ "Запорізька політехніка"

ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА МОВ ПРОГРАМУВАННЯ C++ ТА CLOJURE

Сьогодні незважаючи на появу нових мов програмування, все ще активно використовується така мова як C++. Ця мова прийшла з далекого 1983 року на зміну Common Lisp, Fortran, Algol 68 та інших. Мова C++ – це розширення мови C. При цьому в C++ не намагаються вирішити проблеми шляхом видалення елементів C, а додається можливість роботи з класами та об'єктами. Оригінальна мова C++ була розроблена Б. Страуструпом, для якого базових можливостей звичайного C виявилось замало, коли він почав працювати над задачами теорії черг.

Мова C++ відноситься до об'єктно-орієнтованих мов зі строгою типізацією, можливістю процедурного програмування та імперативною парадигмою. Важливо зауважити, що вона зберігає сумісність з C, наскільки це можливо. Мова C++ з самого початку орієнтована на компіляцію в машинний код заданої платформи (хоча, теоретично, ніщо не заважає створювати для C++ транслятори в проміжний код). Це вже визначає різницю в сферах застосування мов. Мова C++ зберігає можливість роботи з низькорівневими покажчиками та ручним керуванням пам'яттю. З одного боку це можна віднести до переваг, проте на великих проектах це скоріш зайве та заважає сконцентруватися на реалізації потрібного завдання. А от що дійсно завжди супроводжувало C++, так це швидкість компілятора.

У якості протиприкладу, можна розглядати мову Clojure, яка базується на хост-машині Java. У 2007 році Річ Хіккі розробив мову Clojure, як сучасний діалект Lisp, який орієнтується на функціональну складову, та особливо "чисті" функції. Мова Clojure - мова загального призначення, застосовувати його можна для рішення будь-яких задач. Мета, яка ставилася при проектуванні Clojure – зробити написання програми простими. І, як наслідок, прискорити їх створення, тестування. А головне, зменшити час на їх розуміння, редагування і супровід. На мові Clojure можна писати і повноцінні

веб-сайти та виконати досить складні обчислення. При цьому Clojure підтримує інтерактивну розробку за допомогою REPL. Також Clojure полегшує роботу з конкурентним та паралельним програмуванням.

Головною особливістю мови Clojure є імутабельність (незмінність) структур даних. Немає способу змінити елемент вектору. Можливо тільки створити новий вектор, у якого буде змінений один елемент. Дуже важливий момент в тому, що Clojure зберігає алгоритмічну складність (за часом і за пам'яттю) для всіх стандартних операцій над колекціями. Мова Clojure володіє гомоіконністю, тобто написаний код зчитується в структури даних, які виглядають так само як і початковий код. Звідси випливає те, що Clojure має в своєму арсеналі такий потужний інструмент, як макроси. Вони дозволяють виконувати написаний код під час компіляції та підставляти його замість самого макросу вже під час виконання. Якщо використовувати макроси, коли це дійсно потрібно, то можна зменшити обсяг написаного коду. Також варто зауважити, що Clojure підтримує глобально інтерактивну розробку за допомогою середовища програмування REPL (read-eval-print-loop). Ця можливість дозволяє експериментувати з програмою на ходу, тобто без необхідності компілювати її. Clojure нечиста мова, і функції можуть мати побічні ефекти. Наприклад, `println` - це виклик функції, дія. Важливим є те, що сама суть подібних функцій полягає у взаємодії із зовнішнім світом. Вивести значення в файл, відправити HTTP запит, виконати SQL - всі ці дії не мають сенсу у відриві від створюваного ними побічного ефекту. Тому дуже корисно такі функції (чисті і брудні) розділяти.

Особливості мови Clojure: реалізація багатопоточності заснована на незмінності значень і змінних посиланнях на них, які дозволяють управляти паралельним доступом з різних потоків; функції є об'єктами першого класу; підтримка множин, векторів і словників як інтегрованих структур даних; акцент робиться не на цикли, а на рекурсію і функції вищого порядку.

В Clojure добре зроблений інтерфейс взаємодії з колекціями даних і взагалі зі структурами. Всі колекції обробляються однаково за рахунок абстракції «послідовності». Для доступу до елементів послідовностей є багато функцій, які дозволяють отримати бажаний результат незалежно від того, чи працюють вони зі списком, вектором або безліччю.

Однією важливою відмінністю між мовами C++ та Clojure є те, що остання має в своєму розпорядженні потужний інструмент, який автоматично збирає "сміття". При інтенсивній роботі з незмінними колекціями іноді виникає занадто багато проміжних об'єктів, що досить неефективно. В Clojure є можливість тимчасово використовувати змінні колекції даних використовуючи перехідні (transient) структури даних. Ця функціональність була спеціально введена в мову для оптимізації продуктивності. Це стало можливим завдяки Java, на якій базується Clojure. Важливим є те, що Clojure

повністю сумісний з кодом Java, адже у підсумку компілюється у байт-код. Це надає можливість запускати написані на Clojure програми усюди, де підтримується JVM. Однією перевагою виступає також незмінність даних та “чисті” функції, на яких базуються функціонально-орієнтовані мови програмування. Незмінність в Clojure гарантує пряму поведінку даних, адже по суті, всі дані є константами, адже створюються лише один раз і не піддаються зміні. При спробі змінити певну структуру даних, створюється її копія завдяки персистентності. Ця особливість гарантує те, що при створенні, наприклад, нової колекції, вона не копіюється повністю, а копіюється тільки частина, яка була змінена.

Цікавим фактом є те, що замість імперативних циклів у мові Clojure явно використовується декларативне програмування. Це означає вирішення багатьох проблем рекурсивними методами. Особливо гарним стилем є так звана “хвостова” рекурсія, яка в свою чергу не забиває стек виконання.

Мова Clojure – це інструмент робочий, надійний, досить універсальний, простий. Звичайно, для його освоєння доведеться витратити деякий час. Багато що може здатися незвичним та дивним. Але вся краса цієї мови в її органічності, тонкому стикуванні окремих елементів в єдине ціле. Ознайомитися з Clojure однозначно варто. Навіть якщо цей інструмент не пригодиться з тій або іншої причини, то ідеї, які в нього закладені, виявляться дуже корисними.

Таблиця 1 – Порівняння основних концептів та парадигм C++ та Clojure

	C++	Clojure
Семантика	Об’єктно-орієнтована	Функціональна
Типізація	Статична, слабка	Динамічна, сильна
Парадигма	Імперативна	Декларативна
Підтримка гомоіконності	Ні	Так
Очистка сміття	Відсутня	Збирач сміття
Синтаксис	Префіксний. Інфіксний. Постфіксний	Префіксний

Всі мови програмування і технології по-своєму цікаві. І, в принципі, все їх можна задіяти в тій чи іншій сфері. Не можна однозначно сказати, яка мова програмування краще: C++, Java, PHP, Python, Clojure та інші. Вибір технології багато в чому залежить від планів з розвитку продукту та бюджету.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Campbell Parallel Programming with Microsoft® Visual C++® / Campbell. – Москва: Гостехиздат, 2011. – 784 с.
2. Альфред В. Ахо Компиляторы. Принципы, технологии и инструментарий/Альфред В. Ахо и др. – Москва:Высшая школа, 2015.– 882 с.
3. Гонсалвес, Энтони Изучаем Java EE 7 / Энтони Гонсалвес. – М.: Питер, 2016. –640 с.
4. Шилдт, Герберт, Java 8. Руководство для начинающих / Герберт Шилдт. – М.: Вильямс, 2015. – 720 с
5. Daniel Higginbotham, Clojure for the Brave and True: Learn the Ultimate Language and Become a Better Programmer – USA, 2015. – 328 с.

УДК 004.384

Луценко Н.В.¹, Паромова Т.О.¹, Костецкий Д.В.²

¹ старш. викл. НУ "Запорізька політехніка"

² студ. гр. КНТ-518 НУ "Запорізька політехніка"

ЧАТ-БОТИ У СУЧАСНОМУ ІТ-ПРОСТОРИ

У сучасних умовах все більш державних установ та банків надають свої послуги за допомогою Чат-ботів. Чат-бот – це віртуальний онлайн-консультант, який може легко замінити співробітника технічної підтримки, адміністратора, менеджера і навіть продавця, а значить – уберегти бюджет від зайвих витрат, і прискорити роботу [1].

По суті, чат-бот – це сервіс, який веде діалог з Клієнтом за допомогою текстових або голосових повідомлень (команд). Його можна інтегрувати прямо на сайт, але ефективніше використовувати різні месенджери та соціальні мережі [2].

Чат-бот працює постійно. Потенційний клієнт вводить запит і відразу отримує відповідь. Таким чином, здійснюється імітація поведінки живого консультанта в чаті.

Найпоширеніші можливості таких ботів:

- дає відповідь на нескладні питання клієнтів;
- проводить анкетування;
- допомагає продавати;
- опитує працівників;
- бронює номери в готелях і шукає квитки;
- надсилає листи та відповідає на них;
- організовує цілодобову роботу з клієнтами;
- відправляє термінових новин певному колу осіб.

Найчастіше боти використовують для залучення клієнтів, створення комфортних умов для покупки товарів та послуг. Після швидкого отримання необхідної якісної інформації від бота клієнт частіше здійснює покупки або замовляє послуги. Тому розробка та використання цього виду програмного забезпечення є актуальною та затребуваною.

Розробка чат-бота може проводитися за участю програміста і без нього. Перший варіант – найпростіший, реалізується за допомогою спеціальних програм-конструкторів. Вони бувають платні та безкоштовні, все залежить від необхідного функціоналу. При розробці чат-бота можна використати конструктори ChattyPeople, Botify, Chatfuel, BeepBoor.

Другий варіант – складніший і дорожчий, але варто відзначити що чат-бот, написаний програмістом, завжди "розумніший", ніж створений через конструктор. Але програміст-початківець може розробити чат-бот для меседжера за допомогою вбудованих інструментів і знань основ мови Python.

Для месенджера Telegram послідовність створення чат-боту може бути представлена наступним чином:

1. Реєстрація бота.

Треба зайти в месенджер Telegram та викликати вбудований бот @BotFather. Це бот від розробників Telegram, який реєструє нових ботів і дає токен (ключ доступу), який служить для авторизації й взаємодії майбутнього коду з цим ботом. Для отримання цього токена потрібно відправити йому команду start і заповнити поля, які він запитає (назва бота і його коротке ім'я), і в підсумку отримуємо повідомлення з токеном, без якого бот не буде працювати, тому його потрібно зберегти.

2. Підготовка до програмування бота

Встановити Python і середовище розробки PyCharm. Посилання на установку Python та PyCharm наступні:

<https://www.python.org/downloads/windows/>,

<https://www.jetbrains.com/pycharm/download/#section=windows>.

Для встановлення бібліотеки у вікні потрібно прописати "pip install pytelegrambotapi". Після завантаження відображається повідомлення про успішне завантаження.

3. Послідовність програмування.

У будь-якому текстовому редакторі записується код боту. Насамперед потрібно імпортувати бібліотеку встановлену раніше і прописати токен.

```
import telebot;
```

```
bot = telebot.TeleBot('Ваш_токен');
```

Для отримання текстових повідомлень необхідно оголосити слухача та метод їх обробки.:

```
@bot.message_handler(content_types=['text'])
```

```
def get_text_messages(message):
```


Після цього в метод додається такий функціонал:

- Користувач пише *"Привіт"*, то Бот відповідає йому *"Привіт, чим я можу допомогти?"*

- Користувач пише команду *" /help"*, то Бот відповідає *"Напиши Привіт"*

- Користувач пише не те, що описано вище, то Бот відповідає: *"Я тебе не розумію. Напиши /help."*

Отримуємо такий токен:

```
if message.text == " Привіт":  
    bot. send_message(message.from_user.id, "Привіт, чим я можу тобі  
допомогти?")  
elif message.text == " /help":  
    bot. send_message(message.from_user.id, " Напиши Привіт")  
else:  
    bot. send_message(message.from_user.id, "я тебе не розумію. Напиши  
/help.").
```

Для того, щоб Бот постійно перевіряв аутентифікацію користувача, необхідно прописати команду, яка весь час буде опитувати сервер месенджера Telegram.

bot.polling(none_stop= True, interval=0).

І якщо все ж з'являться повідомлення для Боту, то Телеграм-месенджер пришле отримане повідомлення.

Таким чином, створення простішого Боту для Телеграм-месенджеру може стимулювати зацікавленість молоді в засвоєні програмування на мові Python.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. ТОП-10 причин, почему чат-боты в соцсетях нужны каждому [Электронный ресурс]. – Режим доступа: <https://ideadigital.agency/ru/chat-bot/>

2. Как создать и использовать чат-бота с выгодой для вашего бизнеса [Электронный ресурс]. – Режим доступа: <https://geniusmarketing.me/lab/kak-sozdat-i-ispolzovat-chat-bot-s-polzoi-dlya-vashego-biznesa/>

3. 10 этапов разработки качественного чат-бота для клиентской поддержки [Электронный ресурс]. – Режим доступа: <https://chatbotconf.com.ua/ru/article/10-etapov-razrabotki-kachestvennogo-chat-bota-dlya-klientskoy-poddergki-69338>

4. Лутц М. Изучаем Python. / М.Лутц – Пер. с англ. Киев.: ООО «Диалектика», 2020. – 870 с.

5. Реализация Python для Telegram Bot API [Электронный ресурс]. – Режим доступа: <https://pypi.org/project/pyTelegramBotAPI/>

ВИКОРИСТАННЯ СИСТЕМИ DOCKER ПРИ РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Процес розробки програмного продукту складний і дуже важливий у сучасному світі. Залежно від проекту кількість рядків коду може складати від декількох сотень до кількох тисяч (мільйонів) команд. Розробку програмного забезпечення частіше здійснює колектив розробників. Їх рішення та методи реалізації поставленої задачі мають бути узгодженими. Кожен учасник колективу працює над рішенням конкретної задачі. Але проблеми можуть виникати на етапі передачі програмного продукту клієнту, тиражуванні та використання багатьма одного програмного продукту одночасно. Для рішення цієї проблеми була розроблена система Docker.

Система Docker - це програмна платформа для швидкої розробки, тестування і розгортання додатків. Система Docker упаковує ПО в стандартизовані блоки, які називаються контейнерами. Контейнери - це новий рівень віртуалізації IT-інфраструктури. Вони ізолюють окремі процеси всередині однієї ОС і запускають їх з загальним доступом до бібліотек і ресурсів цієї ОС. Завдяки контейнерів кожне окремо запущене застосування зберігає всі переваги хмари: резервування, безперебійність роботи, масштабованість, автоматичне керування. Кожен контейнер включає все необхідне для роботи програми: бібліотеки, системні інструменти, код і середовище виконання. Завдяки системі Docker можна швидко розгорнути і масштабувати додатки в будь-якому середовищі і зберігати впевненість в тому, що код буде працювати. Це не єдина подібна платформа, але, безперечно, одна з найпопулярніших і затребуваних.

Використання системи Docker на AWS (Amazon Web Services) дає розробникам і системним адміністраторам надійний і економічний спосіб збирання, доставки і запуску розподілених додатків будь-якого масштабу. AWS підтримує обидві моделі ліцензування Docker: ліцензію з відкритим кодом Docker Community Edition (CE) і ліцензію Docker Enterprise Edition (EE) на основі передплати.

В основі роботи Docker лежить стандартизований спосіб виконання коду. Docker - це операційна система для контейнерів. Подібно до того, як віртуальна машина створює віртуальне представлення апаратного забезпечення сервера, контейнери створюють віртуальне представлення серверної операційної системи. Після установки на кожен сервер Docker надає доступ

до простих команд, необхідним для збірки, запуску або зупинки контейнерів.

Система Docker відокремлює мережеві ресурси, сховище і деталі ОС. З Docker додаток дійсно не залежить від налаштувань цих низькорівневих ресурсів. При переміщенні Docker-контейнера від одного Docker-хоста до іншої машини з Docker, система гарантує, що навколишнє середовище для додатка залишиться незмінним.

Ця система допомагає програмістам створювати локальні середовища розробки, які виглядають як продакшн-сервер. Коли програміст закінчує писати і починає тестувати код, він може обернути його в контейнер, опублікувати безпосередньо на сервері або в приватній хмарі, і він відразу буде працювати, так як це одне і та ж середовище.

Docker пришвидшує процес розробки програмного забезпечення. В системі не потрібно встановлювати сторонні додатки, їх запускають контейнерах. Docker також дозволяє запускати різні версії одного додатку одночасно. Відділення додатків від апаратного забезпечення, що лежить в їх основі, - фундаментальний концепт віртуалізації. Контейнери йдуть ще далі і відокремлюють додатки від ОС. Завдяки цій особливості програмісти отримують гнучкість і масштабування при розробці.

Переваги використання системи Docker контейнеризації:

- кожен контейнер має невелику вагу і швидко запускається;
- контейнер не є віртуальною машиною і не емулює всю систему;
- контейнер ізолюють один додаток і його залежності - все зовнішні бібліотеки програмного забезпечення, які потрібно запустити з додатком;
- контейнери економлять ресурси всієї системи;
- забезпечує переносимість додатків;
- забезпечує зниження вартості розробки;
- забезпечує стандартизацію роботи додатків;
- забезпечує незалежність від системи - додаток працює ідентично.
- Недоліки системи Docker
 - не вирішує проблеми безпеки самого додатку;
 - не замінює віртуальні машини;
 - не прискорює розробку додатків, робить їх лише більш надійними і ефективними за рахунок посиленої роботи співробітників.
- Отже, система Docker добре підходить для вирішення таких задач:
 - зручна передача серверного проекту клієнтові;
 - забезпечення тиражовані серверів;
 - забезпечення можливості використання раніше створених серверних конфігурацій;
 - зручність поновлення раніше встановлених версій продукту.

Docker є важливим інструментом для кожного сучасного розробника, як основа апаратної віртуалізації додатків. Ця технологія має широкий

функціонал і можливостями для контролю процесів. Docker дозволяє не тільки розгортати контейнери, а й оперативне масштабувати їх екземпляри, працювати з багатоконтейнерними додатками (Docker Compose), а також об'єднувати кілька Docker-хостів в єдиний кластер (Docker Swarm).

Система Docker досить проста в освоєнні як для досвідчених ІТ-фахівців, так і для новачків. Програмне забезпечення сумісне з усіма версіями операційних систем Linux і Windows, тому область застосування Docker практично не обмежена.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Why you should use Docker and containers Serdar Yegulalp [Електронний ресурс]. – Режим доступа: <https://www.infoworld.com/article/3310941/why-you-should-use-docker-and-containers.html>

2. Топ 10 преимуществ, которые вы получите, используя Docker Cloude [Електронний ресурс] – Режим доступа: <https://cloud.kiev.ua/top-10-preimushhestv-kotorye-vy-poluchite/>

3. Контейнеризация [Електронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Контейнерная_виртуализация

4. Что такое контейнер? Обзор технологии контейнеризации [Електронный ресурс]. – Режим доступа: <https://www.cloud4y.ru/about/news/obzor-tekhnologii-konteynerizatsii/>

УДК 658:004

Назаренко О.І.¹, Грушко С.С.²

¹ студ. гр. КНТ-537сп НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

СИСТЕМА ВИМІРЮВАННЯ АТМОСФЕРНИХ ПОКАЗНИКІВ В КОНЦЕПЦІЇ ІНТЕРНЕТА РЕЧЕЙ

Інтернет речі - це новий етап розвитку Інтернету, який значно розширює можливості збору, аналізу і розподілу даних, які людина може перетворити в інформацію і в знання. У цьому сенсі Інтернет речі набувають величезне значення.

Ідеологія Інтернет речей спрямована на підвищення ефективності економіки за рахунок автоматизації процесів в різних сферах діяльності і виключення з них людини.

Концепція Інтернету речей (Internet of Things, IoT) — концепція мережі, що складається із взаємозв'язаних фізичних пристроїв, які мають вбудовані датчики, а також програмне забезпечення, що дозволяє здійснювати

передачу і обмін даними між фізичним світом і комп'ютерними системами, за допомогою використання стандартних протоколів зв'язку.

Були вивчені п'ять ключових областей проблем IoT для визначення найбільш загальних проблем і питань, пов'язаних з цією технологією. Ці області включають безпеку; конфіденційність; інтероперабельність і стандарти; законодавство, нормативні вимоги і права; а також економіку і розвиток.

В якості апаратного забезпечення IoT широко використовують мікроконтролери. Мікроконтролер - це спеціальна мікросхема, призначена для управління різними електронними пристроями. Мікроконтролери вперше з'явилися в тому ж році, що і мікропроцесори загального призначення.

Завдяки різноманіттю мікроконтролерів, мається вибір між різними видами, та різноманіттям характеристик.

Мікроконтролери та платформи на основі яких можна побудувати IoT - це:

- PIC controller;
- Arduino;
- Raspberry pi;
- Atmega8.

У якості основи для IoT була обрана мікросхема Atmega8A. Ця мікросхема доступна за ціною, дає можливість прошивки на мові C у компіляторі AVR CodeVision, та має усі функції, щоб виконати роль ядра у IoT для обробки атмосферних показників. Задля матеріального проектування IoT були використані датчики – датчик вологості та температури DHT-11, датчик атмосферного тиску і температури BMP180 та Wi-Fi модуль ESP8266 12-f.

Atmega8A - малопотужний 8-розрядний КМОП мікроконтролер, заснований на розширеній AVR RISC - архітектурі. За рахунок виконання більшості інструкцій за один машинний цикл ATmega8 досягає продуктивності 1 млн. операцій в секунду, що дозволяє проектувальникам систем оптимізувати співвідношення енергоспоживання і швидкодії.

Очікуваним результатом роботи має бути «домашня метеостанція» яка передає свої показники до мережі Інтернет, що в свою чергу передає данні на хмару.

Таке рішення повністю відповідає концепції Інтернету речей та може бути використане у глобальній схемі «Розумний дім». Розроблювана система містить датчик температури, тиску, висоти та Wi-Fi модуль, завдяки якому можна передавати данні до глобальної мережі Інтернет.

Ідея розробки схеми була передача температурних показників з датчиків на сервер, а після цього на смартфон користувача. У часи Інтернет речей така система має вагоме місце, тому що знання точної погоди в окремому місці,

інформацію про яку можна подивитися на смартфоні дуже полегшує життя та економить час.

В подальшому таку систему можна використовувати у тандемі з іншими IoT та створити на їх основі систему, що забезпечує концепцію розумного дому.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Скаржепа В.А. Электроника и микросхемотехника. Ч.1. Электронные устройства информационной автоматики / В.А. Скаржепа, А.Н. Луценко – К.: Выща шк., 1989. – 431 с.

2. Прищепа М.М. Мікроелектроніка. Ч.1. Елементи мікроелектроніки / М.М. Прищепа, В.П. Погребняк. – Київ: Вища школа, 2004. – 432 с.

3. Як розумні контракти змінюють наше життя [Електронний ресурс]. – Режим доступу: <https://habrahabr.ru/company/exante/blog/321548/> -

4. Забродин Ю.С. Промышленная электроника / Ю.С. Забродин. – М.: ВШ, 1982. – 496 с.

УДК 004.514

Наливайко А.А.¹, Тягунова М.Ю.²

¹студ. гр. КНТ-519м НУ «Запорізька політехніка»

²канд. техн. наук, доц. НУ «Запорізька політехніка»

МЕТОДИКА ПРОЕКТУВАННЯ ГРАФІЧНОГО ІНТЕРФЕЙСУ МОБІЛЬНОГО ДОДАТКА

Індустрія мобільних телефонів вважається однією з найбільш прогресивних в сучасному світі. Більше 60% населення планети, незалежно від віку та професії, щодня користуються смартфонами [1].

В Україні ж, на середину 2019 року, ця частка становила 55%, тоді як ще в 2011 році користувачів смартфонів було всього 7% [2]. У зв'язку з цим стрімко розвивається і ринок розробки мобільних додатків. Для забезпечення зручностей у будь-якій сфері діяльності і життя людини з'являються все нові і нові додатки: інтернет-магазини, доставки їжі, квітів, мобільні банки тощо. Конкуренція зростає із геометричною прогресією, а основні задумки вже давно запатентовані, проте розробка мобільних додатків все ж залишається прибутковою галуззю і продовжує розвиватися. Саме через ситуацію, що склалася, при розробці мобільних додатків все більше часу і ресурсів витрачається саме на етап проектування графічного інтерфейсу.

Графічний інтерфейс – це не тільки створення іконок та кнопок, визначення розмірів елементів, кольорів та шрифтів. Це ще довгий і кропіткий процес дослідження цільової аудиторії, для якої розробляється

додаток, занурення в її навколишнє середовище, співпереживання користувачам і розуміння їх потреб та проблем, пошук рішень та розробка, в першу чергу, зручного, а тільки потім естетичного графічного інтерфейсу. Щоб виділитися серед конкурентів, додаток повинен мати інтуїтивно зрозумілий і лаконічний інтерфейс, адже саме він стоїть між користувачем і функціоналом додатку. Тільки дивлячись на графічний інтерфейс користувач відразу може визначити, залишить він додаток на своєму смартфоні, або ж одразу видалить його.

Етапів розробки графічного інтерфейсу запропоновано незліченну кількість: кожен UI/UX дизайнер обирає свою послідовність кроків, за якою йому зручно працювати. Деякі фахівці починають розробку з аналізу цільової аудиторії, а деякі – з аналізу конкурентів. Перші можуть почати із занурення в предметну область, інші – відразу приступити до створення ескізів екранів та визначення кольорової палітри.

Існують базові теми, які повинні бути в обов'язковому порядку розкриті в процесі роботи над графічним інтерфейсом мобільного додатка: дизайн-дослідження, визначення функціоналу додатка, розробка сценаріїв взаємодії з користувачем, а слідом і прототипів всіх екранів, визначення кольорової палітри та відповідних тематиці шрифтів, розробка візуальної частини дизайну та його тестування одним або декількома існуючими методами. І кожна з цих тем може розгалужуватися на десятки підпунктів.

Метою цієї роботи є розробка універсальної покрокової методики проектування графічного інтерфейсу мобільного додатка. Розроблена методика містить не тільки всі необхідні етапи, а й аргументацію кожного вибору, приклад проектування графічного інтерфейсу за нею, а також апробацію та доказ її універсальності.

Процес розробки методики починається з аналізу існуючих аналогів, застосовуваних великими ІТ-компаніями і дизайн-студіями, адже саме вони є прикладами для наслідування. Детальну інформацію з прикладами вони надати не можуть, бо вона закрита корпоративним договором, але основний список кроків визначити можливо. В результаті аналізу виявлено, які загальні етапи проектування графічного інтерфейсу проходять обрані організації. Отже, ці етапи і є основними в усьому процесі.

На наступному етапі розробки необхідно проаналізувати, які розгалуження є у кожного основного кроку, визначеного раніше, зазначити їх переваги і недоліки, їхню необхідність в усьому процесі, а також аргументувати вибір тих чи інших пунктів.

Результатом роботи є докладна покрокова методика, універсальна для проектування більшості графічних інтерфейсів мобільних додатків.

Для доказу заявленої універсальності розробленої методики, вона була випробувана для проектування графічного інтерфейсу мобільного додатка

для медичних працівників – Patient Manager. Сфера медицини обрана тому, що ця ніша в державі ще вільна, аналогів подібних додатків всього кілька і навіть вони мають обмежений функціонал. Основними вимогами є організація зручного управління записами на прийом для медичних працівників, а також відстеження фінансового потоку медичних працівників і можливість запису на прийом.

Для аргументації зручності та підтвердження працездатності методики спроектований графічний інтерфейс було протестовано серед групи людей, які відповідають цільової аудиторії, та оцінено ними за певними критеріями. Результати тестування підтвердили очікуванні сподівання.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Юдін А. Найпопулярніші виробники смартфонів 2019 / А. Юдін [Електронний ресурс]. – Режим доступу: <https://marketer.ua/mobile-device-provider-market-2019/>

2. Страна UA. Новини за 2019 рік. Інфографіка [Електронний ресурс]. – Режим доступу: <https://strana.ua/news/193560-v-ukraine-za-poslednie-pjat-let-v-pjat-raz-vyroslo-chislo-vladeltsev-smartfonov-infohrafika.html>

УДК 004.42

Нестюрич С.І.¹, Точилін С.Д.²

¹студ. гр. КНТ-519м НУ «Запорізька політехніка»

²канд. фіз.-мат. наук, доц. НУ «Запорізька політехніка»

WEB-ДОДАТОК - ЛАБОРАТОРНА РОБОТА «ВИЗНАЧЕННЯ КОЕФІЦІЄНТА В'ЯЗКОСТІ РІДИНИ»

У наш час навчальні заклади розміщують в мережі Internet навчальні матеріали та навчальні комп'ютерні програми. У цьому зв'язку розробка навчальних Web-додатків є актуальною задачею.

У даній роботі розроблено Web-додаток - лабораторна робота «Визначення коефіцієнта в'язкості рідини», який розміщався на Web-сервері. Він складався з html-сторінки Ligature.html та вбудованої в цю сторінку програми ST_Ligature (Java-апплет).

Вікно браузера при виклику html-сторінки Ligature.html, яка містить апплет ST_Ligature, зображене на рис.1.

Програма ST_Ligature має графічний інтерфейс користувача. На початку виконання роботи студент за допомогою повзунка вибирає свій варіант. Після натискання на кнопку «Run» починається виконання експериментальної частини роботи - кулька починає рухатися і досягає дна

посудини с рідиною. Час її руху між міткою та дном посудини фіксується таймером. Кнопка «Reset» приводить додаток у вихідний стан.

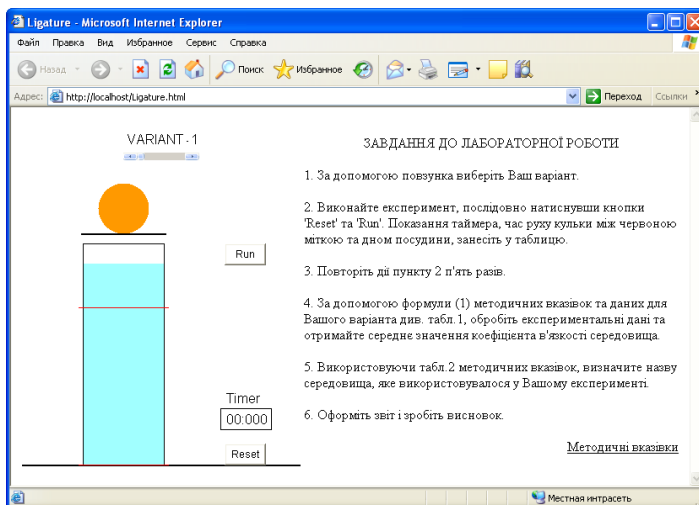


Рисунок 1 - Вікно браузера з html-сторінкою Ligature.html

На рис. 2 зображене вікно браузера при виконанні лабораторної роботи (обрано варіант № 3).

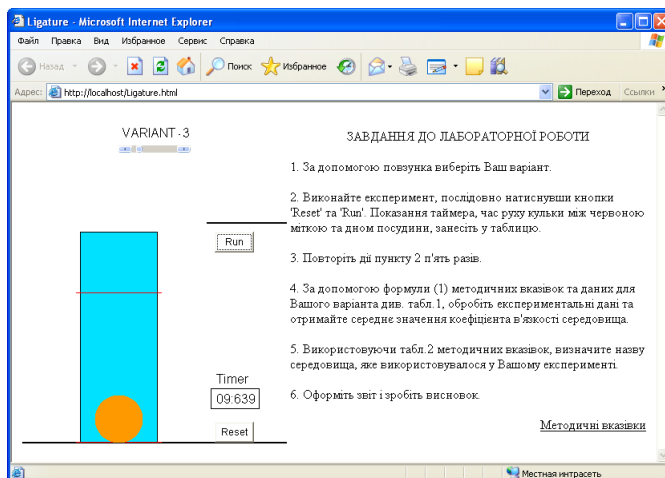


Рисунок 2 - Вікно браузера при виконанні лабораторної роботи

Розроблений Web-додаток може працювати, як у мережі Internet, так і в мережі intranet, а також на локальному комп'ютері.

Надалі передбачається модернізувати програму ST_Ligature, зокрема, забезпечити можливість збереження результатів виконання лабораторної роботи.

УДК 004.43

Предко В.Ю.¹, Киричек Г.Г.²

¹ студ. гр. КНТ-516 НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

МІКРОСЕРВІСИ ПРОТИ МОНОЛІТІВ

Монолітний підхід є найстарішою моделлю проектування програмного забезпечення (ПЗ). В рамках даної архітектури веб-додатки не мають складної структури: сервер зберігає бізнес-логіку, а база даних (БД) – необхідні для роботи сервера дані. Ця архітектура формує додаток у вигляді пов'язаного модуля, де всі компоненти працюють разом, розділяючи спільну пам'ять і ресурси (рис.1). Подібні програми не є складними в розробці і мають низку вартість на ранніх етапах розробки. Однак, при допущенні помилок, маємо проблеми які складно виправити після розгортання продукту. При цьому таких помилок може накопичитися дуже багато, тому підтримувати систему монолітної архітектури в довгостроковій перспективі складно і дорого, іноді простіше зробити «hotfix», ніж розібратися як працює частина програми. Тому проблеми з підтримкою монолітних додатків можна пояснити тим, що в майбутньому весь список функціоналу системи буде потребувати модернізації та внесення змін до вже існуючого коду. Відсутність масштабованості в монолітній архітектурі, коли всі складові системи розташовані в одній точці, веде до того, що потужність цієї точки повинна бути придатною для підтримки в робочому стані сервера і бази даних [1].

Мета роботи – дослідження можливостей масштабування та впровадження автономності функціональних команд і модулів програмного забезпечення та створення архітектури високого ступеня деталізації.

Об'єкт дослідження – процес реалізації мікросервісної архітектури.

Мікросервісна архітектура – підхід до створення додатка, що виключає використання єдиної та монолітної структури. Тобто замість того, щоб виконувати всі контексти додатка на сервері за допомогою внутрішніх процесів взаємодії, використовується декілька додатків, кожен з яких відповідає своєму обмеженому контексту [2].

Сервіси є окремими цілком самодостатніми модулями, які мають власну апаратну базу та розташовані на окремому сервері. Також вони використовують власну базу даних, можуть розміщуватися на окремих апаратних пристроях та, навіть, якщо і віртуально, взаємодія між сервісами здійснюється асинхронно, що надає як певні переваги так і недоліки (рис.1) [1].

Недолік монолітної архітектури у тому, що її складно швидко змінювати і адаптувати в ній великі сервіси. Нескінченно великий додаток з нескінченною функціональністю монолітним бути не може. Для великих додатків корпоративного масштабу використання мікросервісної архітектури є оптимальним рішенням, яке дозволяє швидко реагувати на зміни, що відбуваються на ринку [3].

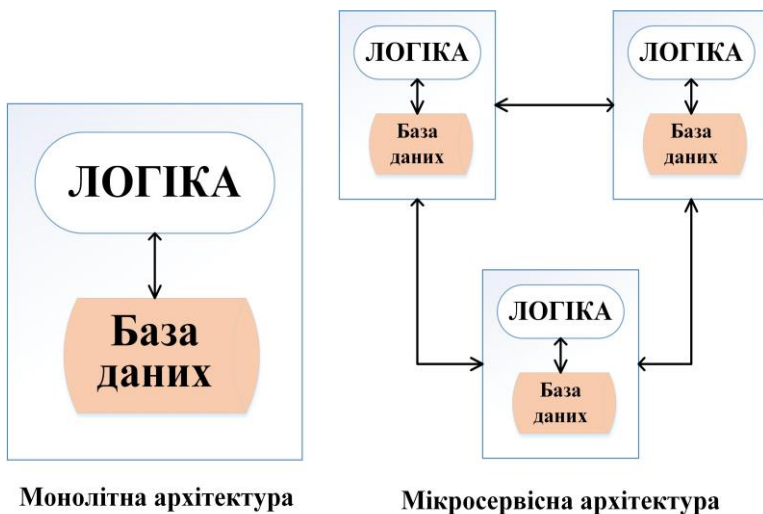


Рисунок 1 – Схема монолітної та мікросервісної архітектури

В процесі реалізації окремі функціональні модулі програм запускаються на різних серверах або різними процесами. Тому, в процесі розробки програмного забезпечення, необхідно змінювати стратегію самої розробки та розділяти додатки на окремі модулі, які реалізують конкретний функціонал або вирішують окремі завдання. Такими модулями є сервіси, які відповідають за конкретний функціонал. Наприклад, завантаження файлів виконує окремий сервіс, а інші виконують свої дії нічого не знаючи про завантаження файлів.

Мікросервіси є окремими монолітними програмами, функції яких

обмежені власними завданнями, що не впливають на стан інших завдань. Програмний продукт з платформою мікросервісної архітектури може включати будь-яку кількість нових мікросервісів, розширюючи свою функціональність [3]. Простота та швидкість впровадження мікросервісів зумовлена тим, що необхідні функції знаходяться в межах одного модуля, тому при виникненні помилок їх відразу ж відслідковують при тестуванні та ліквідують. При внесенні змін до одного сервісу, інші функціональні модулі системи залишаються незмінними: досить просто виключити непотрібний модуль без впливу на весь додаток або додати новий. Це спрощує реалізацію систем в цілому та проведення їх тестування.

Практичне застосування мікросервісної архітектури, більш ніж моноліт забезпечує надійний захист корпоративних даних та даних користувачів. Це пов'язано з тим, що додатки не мають єдиної бази даних, яка б зберігала всю важливу інформацію, бо для кожної частини програми використовується власний простір інформаційних даних.

Зазначимо, що використання одразу декількох БД в одному модулі є допустимим і широко використовується в сучасних розробках. Наприклад, для зберігання конфіденційної інформації користувача, використовується не менше двох БД – одна для публічних даних (ім'я, рік народження, адреса електронної пошти, тощо), а друга для приватних даних (логін, хеш паролю, секретне питання, тощо). При цьому до них може підключитися будь-яка система управління базами даних.

Одна з головних переваг мікросервісної архітектури полягає в тому, що кожен модуль є окремою структурою і може, для реалізації своєї функціональності, використовувати будь-яку програмну платформу. Це означає, що його можна реалізувати будь-якою мовою програмування, тією яка на даний час є найбільш ефективною для реалізації необхідного функціоналу. Однак, на практиці, проблемою є нестача фахівців, які можуть підтримувати весь спектр технологій, що використовуються.

Беручи до уваги всі переваги та недоліки даних архітектур, перебудова або перенесення сервісу з моноліту на мікросервісну архітектуру не завжди є доречним. Якщо програмне забезпечення не передбачає подальшого розширення або воно вирішує незначне коло завдань, немає сенсу вносити в ПЗ будь-які зміни. Але, якщо додаток з великою кількістю об'ємних та різноманітних завдань вже працює з підтримкою монолітної архітектури, то маючи всі необхідні ресурси для капітального оновлення системи, потрібно не до-опрацьовувати новий функціонал у межах старого, а повністю його оновити змінюючи підхід та впроваджуючи мікросервісну архітектуру.

В роботі розглянуто дві моделі проектування програмного забезпечення та реалізації функціональних команд і модулів – мікросервісну та монолітну. Визначені головні переваги мікросервісної архітектури перед монолітною та

зроблені висновки у яких випадках доречно переходити на архітектуру з високим ступенем деталізації, а у яких ні.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Богданенко Д. А. Подходы к архитектурному проектированию веб-приложений / Д. А. Богданенко [Електронний ресурс]. – Режим доступу: <https://moluch.ru/archive/195/48609/>.

2. Осипов Д.Б. Проектирование программного обеспечения с помощью микросервисной архитектуры / Д.Б. Осипов [Електронний ресурс]. – Режим доступу: <https://cyberleninka.ru/article/n/proekti-rovanie-programmnogo-obespecheniya-s-pomoschyu-mikroservisnoy-arhitektury>.

3. Ньюмен С. Создание микросервисов. / С. Ньюмен. – СПб: Издательский дом "Питер", 2016. – 307с.

УДК 004.4; 65.01

Предко В.Ю.¹, Польська О.В.²

¹ студ. гр. КНТ-516 НУ «Запорізька політехніка»

² старш. викл. НУ «Запорізька політехніка»

РОЛЬ ТАЙМ-МЕНЕДЖМЕНТУ В ОРГАНІЗАЦІЇ ІТ-ПРОЕКТУ

Оцінка часу розробки проекту – одна з важливих задач в ІТ-індустрії. Час тривалості проекту встановлюється, коли проект знаходиться ще в стадії «концепції» до або без складання докладного графіку всіх операцій проекту.

Завжди існують ризики несвоєчасного виконання проекту, наприклад непередбачувані затримки (перерозподіл виконавців, недоліки проекту, проблеми з обладнанням тощо) [1]. При цьому важлива роль приділяється тайм-менеджменту, або управлінню часом. В загальному розумінні це дія або процес тренування свідомого контролю над часом, витраченим на конкретні види діяльності, при якому спеціально збільшуються ефективність і продуктивність. Виділяють два види тайм-менеджменту: особистий і корпоративний.

В управлінні часом можна виділити наступні процеси: аналіз, моделювання стратегій з урахуванням проведеного аналізу, цілепокладання, планування і розстановка пріоритетів, реалізація, контроль досягнення мети.

Критерії розподілення пріоритетів в тайм-менеджменті можуть бути різними - це і особисті мотиви, і фінансова винагорода. При розстановці пріоритетів користуються наступними методиками планування [2]:

- за системою АВС, де завдання розподіляються за ступенем значущості за показником терміновості;

- за принципом Ейзенхауера, де пріоритети розподіляються за категоріями, це дозволяє виділити важливі і суттєві задачі і вирішити, що робити з рештою;

- за принципом Парето, емпіричне правило якого в найбільш загальному вигляді формулюється як «20% зусиль дають 80% результату»;

- за технологією постановки і формулювання цілей SMART (конкретні, вимірювані, досяжні, значущі, з часовим обмеженням).

Отже, організація ефективного тайм-менеджменту сприятиме скороченню часу розробки в умовах ризику та часових обмежень ІТ-проекту.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Катренко А.В. Управління ІТ-проектами : підручник /А.В. Катренко – Львів : «Новий світ-2000», 2011. – 550 с.

2. Тайм-менеджмент для ІТ-спеціалістів [Електронний ресурс]. – Режим доступу: <https://dou.ua/lenta/articles/time-management>

УДК 004.353

Сгадов С. О.¹, Тур'янський Е.Д.²

¹ старш. викл. НУ "Запорізька політехніка"

² студ. гр. КНТ-516 НУ "Запорізька політехніка"

ОБ'ЄКТНО-ОРІЄНТОВАНИЙ ПІДХІД ДО ДИЗАЙНУ ІНТЕРФЕЙСІВ МІКРОПРОЦЕСОРНИХ ПРИСТРОЇВ

При розробці пристроїв на мікроконтролерах широко використовуються пристрої для виводу графічної інформації такі як LCD, TFT, OLED екрани з невеликою діагоналлю порядку 1" - 6". Незважаючи на невеликий розмір ці екрани представляють досить великі можливості по відображенню графічної й текстової інформації. Наприклад, 2.2" TFT екран на драйвері ILI9341 має вирішення 240x320 з можливістю виводу 16-бітного кольору [1]. Для таких екранів розроблені бібліотеки низького рівня, наприклад, на LCD Wiki [1], які дозволяють установлювати колір пікселів і малювати графічні примітиви. Так само нескладно знайти бібліотеки для створення й виводу растрових шрифтів [2]. Проте актуальною є завдання створення й підтримки зовнішнього вигляду екрана додатка в runtime, особливо якщо вивід того самого елемента екрана відбувається в різних позначках програми. Щоб зробити створення й підтримку екранів мікроконтролерного пристрою нами була розроблена система класів мовою C++ для вистави екрана (Малюнок 1). Базовим пристроєм був обраний мікропроцесор STM32F405VE.

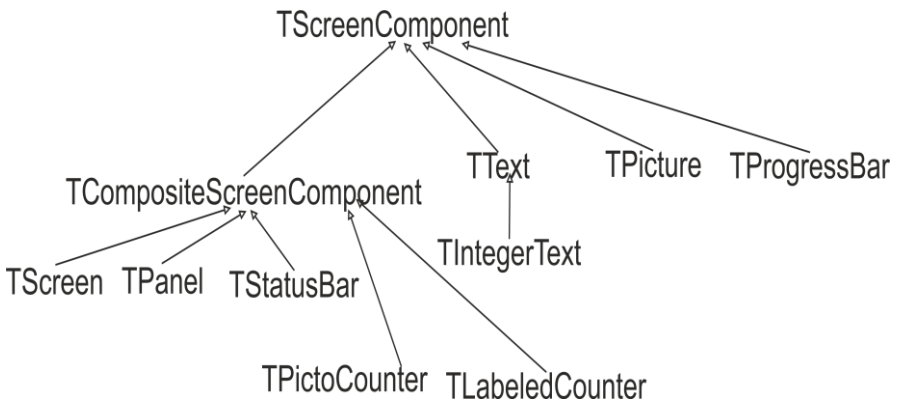


Рисунок 1 – Система класів для вистави екрана

Загальним предком для всіх класів є **TScreenComponent**. Він має координати, розміри, видимість і власника. Його основними методами є: віртуальний метод **Draw()**, який робить отрисовку компонент, віртуальний метод **Clear()**, який відповідає за очищення області компонента. Передбачається, що вони повинні бути перевизначені в спадкоємцях. Наприклад, його спадкоємці **TText**, **TPicture**, **TProgressBar** - відповідно відображають текст, картинку, прогресбар.

Наступним важливим класом є абстрактний клас складеного компонента **Tcompositescreencomponent**, спадкоємець якого може бути власником декількох інших компонентів. Для цього він має член **Components [MAX_SCREEN_COMPONENTS]** ; і метод **Add(TScreenComponent* Component)**. Більше того, якщо конструктор **TScreenComponent** ухвалює ненульовий показник на **Owner**, те він намагається додати свій екземпляр у колекцію **TCompositeScreenComponent**. Це дозволяє конструювати екран або компоненти екрана на етапі конструювання класів. Наприклад, клас **TScreen** представляє абстракцію екрана і його конструктор викликає процедуру ініціалізації дисплея.

Наступним класом є **TPanel**. Основною ідеєю введення цього компонента є не тільки представлення частини екрана, яка володіє іншими компонентами, але й можливість миттєво перемикає вміст екрана, роблячи одну панель видимою, а іншу невидимою.

Наприклад наступний екран складається з рядка статусу, панелі **Panel1**, і панелі **Panel2**. **TPanel1** і **TPanel2** є спадкоємцями **TPanel** та припустимо володіють різними елементами виводу похідними від **TScreenComponent**.

Лістинг 1 – Опис класу TMyScreen

```
class TMyScreen:public TScreen
{
public:
    uint16_t bgcolor;
    TStatusBar1 StatusBar;
    TPanel1 Panel1;
    TPanel2 Panel2;
    TMyScreen(uint16_t bcolor):TScreen(),
        StatusBar(this), Panel1(0,50,bcolor,this), Panel2(0,50,bcolor,this)
    {}
}
```

Тоді можна перемкнути вистави викликом `Panel1.Setvisible(false);`
`Panel1.Setvisible(true);`

Висновки. У даній роботі ми розробили об'єктний підхід для дизайну й виводу інформації для мікропроцесорних пристроїв. Цей підхід дуже близький до об'єктному підходу IDE типу Delphi, отже, можна розвинути його надалі в Visivig інструмент для дизайну інтерфейсів вбудованих пристроїв з використанням натівного коду на C++.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. 2.2inchSPI ModuleILI9341SKU:MSP2202[Електронний ресурс]–Режим доступу:http://www.lcdwiki.com/2.2inch_SPI_Module_ILI9341_SKU:MSP2202

2. Arduino і Android [Амперка / Вики] [Електронний ресурс] – Режим доступу: wiki.amperka.ru/android-и-bluetooth

УДК 004.822

Стусь С.В.¹, Тягунова М.Ю.²

¹ студ. гр. КНТ-519м НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

АКТУАЛЬНІСТЬ СУЧАСНИХ CMS СИСТЕМ

Для спрощення процесу роботи з сайтом та підвищення ефективності фінансових, та трудових витрат розроблені системи управління контентом, які використовують принципи роботи динамічних сайтів.

CMS (з англ. Content Management System) – система управління

контентом, що являє собою програмний комплекс, автоматизує процес розміщення матеріалів, комунікаційну взаємодію з користувачами [1].

Головною перевагою CMS є легкість в управлінні сайтом, відсутність потреби від адміністратора глибоких технічних знань веб-технологій. Кожна функція традиційно виконується окремим програмним модулем, написаним на скриптовій мові.

На сьогоднішній день існує багато таких систем: WordPress, Joomla, 1С-Бітрікс, OpenCart, Drupal, Wix та інші.

З усієї величезної маси веб-сайтів, що наповнюють Інтернет, приблизно 31% використовують CMS. Нижче наведено графік, що показує історичну динаміку долі веб-сайтів, починаючи з 2010 року [2].

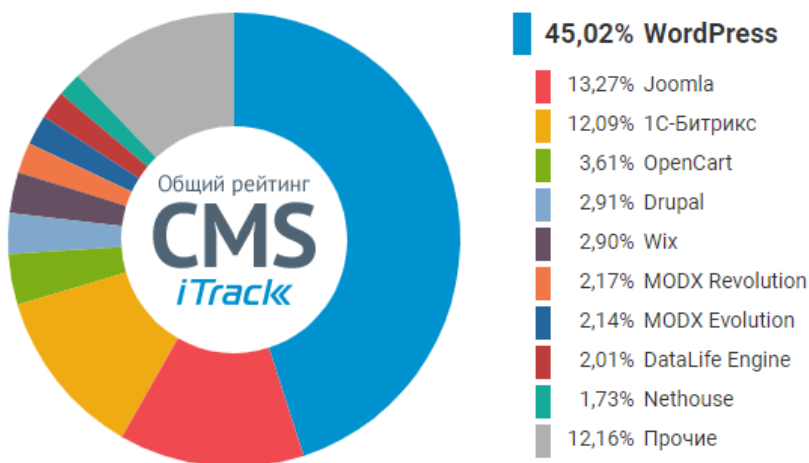


Рисунок 1 – Загальний рейтинг CMS

Даний рейтинг складено за інформацією про реальні дані з сайтів та визнано достовірним більшістю розробників CMS та перевірено компанією Microsoft.

Графік показує виразну позитивну тенденцію в сторону збільшення долі сайтів з CMS, а все тому, що ці інструменти направлені на спрощення керування людиною. Існує безліч CMS, і ви можете вибрати найбільш необхідну для себе, в залежності від цілей, коштів та специфіки вашого веб-сайту. Нижче наведена таблиця характеристики актуальних CMS-систем.

Таблиця 1 – Коротка характеристика актуальних CMS-систем

	Характеристика			
	Призначення	Вартість	Переваги та недоліки	Дизайн
1	2	3	4	5
WordPress	Створення різних сайтів, від лендінгів до багатосторінкових порталів з мільйонною аудиторією.	Безкоштовна.	Найбільш популярна CMS, більша частина користувачів віддають перевагу саме їй. Головна її проблема як використовувати усі її можливості правильно.	Існує велика кількість шаблонів, від простішого портфоліо з кількома сторінками до багатофункціонального інформаційного порталу.
Joomla	Створення інтернет-магазину або сайтів іншої направленості.	Безкоштовна.	Joomla – це гнучкість, простий процес установки для недосвідчених користувачів, підтримка протоколів контролю доступу (OpenID, LDAP, Gmail.com). Недоліком є високий процент зламаних сайтів, довге завантаження сайтів.	Є декілька варіантів вибору дизайну: Завантаження безкоштовних шаблонів для Joomla. Покупка преміум-тем. Самостійне створення макету. Замовлення розробки індивідуального рішення. Єдина складність – такі теми можуть не підтримувати українську мову, але це виправляється редагуванням файлів [3]
1С-Бітрікс	1С-Бітрікс добре структурований, призначений для професіоналів та високобюджетних проектів.	1800-2400 грн.	Високий рівень безпеки та підтримки програм 1С, а з іншого боку досить дорогий, порівняно важкий та громіздкий.	Сайти створюються на основі шаблонів, які визначають їх структуру та способи представлення контенту. Усі шаблони мають адаптивну верстку, що забезпечує відмінне відображення контенту на різних пристроях.

Продовження таблиці 1

1	2	3	4	5
OpenCart	Створення інтернет-магазинів для малих та середніх бізнесів.	Безкоштовна.	Перевагами є імпорт в Excel, відкритий код, простий інтерфейс, велика кількість розширень, просунуте суспільство. Серйозним недоліком є «ЛПУ» (людино-подібний урл) – проблеми з кодом 404. Не повністю налагоджена робота з SEO.	Є якісні безкоштовні теми та преміум-шаблони – вибір залежить від того, що шукати та які засоби є в розпорядженні. Єдина універсальна рекомендація – брати макети з адаптивною версткою. Візуального редактору для внесення змін в інтерфейсі немає, тому прийдеться або задовольнятися тим внутрішнім видом, який пропонує шаблон, або правити його файли [4].
Drupal	Створення великих-інтернет магазинів, система для професіоналів.	Безкоштовна.	Відмінна SEO-оптимізація, є можливість створювати свої типи контенту. Складна в освоєнні, категорично не відходить для новачків.	Редагування теми виконується двома основними інструментами: Block Layout — візуальний редактор, який пропонує такі самі інструменти, як конструктори. Theme Settings — базові параметри теми, в яких можна змінити кольорову схему, налаштувати відображення картинок, задати лого та фавікон. Третій спосіб зміни внутрішнього виду – редагування файлів шаблону [5].

З усього вищенаведеного (див. табл. 1) можливо зробити висновок, що існуючі популярні CMS поділяються на два види: платні та безкоштовні. Кожна з них має окремі недоліки та переваги. Основними перевагами безкоштовних CMS є безкоштовність; велика кількість компаній, приватних осіб, фрілансерів, що можуть надати допомогу; розширення на достатньо високому рівні. До недоліків відноситься: відсутність офіційної підтримки, необхідність спеціальних знань для налаштування систем, відсутність документації. Тож доречно сказати, що вибір CMS залежить від цілі використання, технічних вимог та бюджету.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Рейтинг CMS: CMS Resources [Електронний ресурс]. – Режим доступу: ru.wikipedia.org/wiki/Система_управления_содержимым.
2. Рейтинг CMS: CMS Resources [Електронний ресурс]. – Режим доступу: <https://itrack.ru/research/cmsrate>.
3. Норт, Бэрри Joomla! 1.5. Практическое руководство / Бэрри Норт. – М.: Символ-плюс, 2017. – 464 с.
4. OpenCart преимущества и недостатки: CMS Resources [Електронний ресурс]. – Режим доступу: <https://webbuilding.com.ua/articles/opencart-advantages-disadvantages>.
5. Обзор CMS Drupal: CMS Resources [Електронний ресурс]. – Режим доступу: <https://konstruktorysajtov.com/cms/drupal>.

УДК 004.9

Твердохліб М.Ю.¹, Киричек Г.Г.²

¹ студ. гр. КНТ-526 НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

СИСТЕМА З ПІТРИМКОЮ ТЕРМІНАЛУ ОБСЛУГОВУВАННЯ

Діджиталізація стрімко проникає в усі куточки нашого життя, визначаючи те, як ми проводимо свій вільний час, як будуємо бізнес та наскільки швидко працює сфера обслуговування. Цей термін стає обов'язковою частиною стратегії бізнесу, адже цифрові технології проникають все активніше в усі сфери життя і охоплюють всі верстви населення. Для успішної роботи в умовах сучасного ринку, заклади харчування повинні забезпечити споживачеві швидкий сервіс і якісну продукцію, розуміючи, хто є споживачем закладу, які послуги користуються попитом та наскільки високий рівень обслуговування клієнтів [1].

Метою роботи є реалізація методів та технологій для автоматизації

сфери послуг. Об'єкт дослідження – процес реалізації системи термінального обслуговування клієнтів закладів харчування. Предмет дослідження – моделі, методи, програмні та інструментальні засоби реалізації подібних систем.

Для досягнення поставленої мети потрібно вирішити наступні завдання: дослідити аналоги подібних систем та провести аналіз їх недоліків; змодельовати систему термінального обслуговування клієнтів та реалізувати її загальні модулі, а також прискорити час обробки замовлень; полегшити роботу сфери обслуговування, шляхом надійної фіксації складу і вартості страв за допомогою реалізованого додатку; знизити час перебування персоналу на місті приготування страв.

Система має ієрархічну архітектуру та складається з наступних модулів: android додаток; сервер; сегменти wi-fi мережі; пристрої з встановленим додатком, які розміщені в містах знаходження персоналу та на всіх точках обслуговування клієнтів [2].

Додаток для офіціантів, сучасна заміна паперовим блокнотам офіціанта. Він включає в себе базу даних, під управлінням MySQL, з найменуванням, описом та вартістю страв і напоїв. СУБД MySQL проста в роботі, дозволяє створювати розподілені операції для різних користувачів чий обов'язки і права доступу відрізняються. Вхід в додаток здійснюється за допомогою логіну та паролю за якими закріплена відповідна роль [3].

Сучасні програмні рішення дозволяють прискорити процес прийому замовлення, замінивши блокнот офіціанта на планшет або смартфон і POS-термінал.

POS-термінал компанії xPOS@Ho.Re.Ca. сучасна заміна паперовим блокнотам офіціанта і повноцінний касовий апарат [1]. Пристрій дозволяє створювати замовлення, передавати його на кухню, виконувати розрахунок з клієнтами. У програмному забезпеченні xPOS SYSTEM є POS-термінал офіціанта та xPOS Kassa що відображає всю інформацію по рахунках клієнта.

Робочі місця персоналу оснащуються сучасними планшетами з підтримкою бездротових з'єднань [4]. Під час формування замовлення офіціант вибирає усі його компоненти з бази даних, формуючи "повідомлення" на кухню, яке відправляється відразу після підтвердження замовлення. Повідомлення приходить у вигляді списку з часом замовлення, найменуванням і кількістю страв. Коли замовлення готує кухар натискає на повідомлення в списку і офіціант отримує сигнал про готовність страви. Повністю виконане замовлення видаляється з пристроїв і копіюється на сервер, з метою подальшого аналізу.

Додаток офіціанта реалізовано на мові програмування Java для платформи Android з підтримкою Android Studio [2]. За допомогою віртуальної машини Java програми на ній запускаються практично в будь-якій системі.

Переваги системи: підвищення лояльності користувачів за рахунок швидкого сервісу, миттєвої передачі замовлення та спритності офіціантів; зменшення часу замовлення надає додатковий час на приготування страв, що позитивно позначається на їх якості; усунення помилок розрахунку, виключає наявність забутих замовлень (усе відбувається в режимі реального часу); підвищення продуктивності обслуговуючого персоналу, шляхом вивільнення часу для виконання додаткових завдань.

Система дозволяє оптимізувати процес обслуговування за рахунок автоматизації роботи персоналу закладів харчування. Оснащення робочих місць сучасними планшетами з підтримкою бездротових з'єднань та реалізованими активними додатками, знижують часові затрати та усувають ряд помилок при обслуговуванні клієнтів. Це в свою чергу дозволяє підвищити загальний рівень та категорію закладу, забезпечуючи його стабільний розвиток та збільшуючи здобутки.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Дарвин Ян Ф. Android. Сборник рецептов. / Ян Ф. Дарвин – М.: Вильямс, 2018. – 768с.
2. Гриффитс Д. Head First. Программирование на Android. / Д. Гриффитс, Д. Гриффитс – СПб.: Питер, 2018. – 912с.
3. Kirichek G. Decentralized System for Run Services / Kirichek G., Tymoshenko V., Rudkovskyi O., Hrushko S. // CMIS. – 2019. – pp.860-872.
4. Олифер В. Компьютерные сети. Принципы, технологии, протоколы / В. Олифер, Н. Олифер. – СПб.: Питер, 2016. – 992с.

УДК 004.67

Тіменко А.В.¹, Бубенець О.В.²

¹ асист. НУ "Запорізька політехніка"

² студ. гр. КНТ-519м НУ "Запорізька політехніка"

ВИКОРИСТАННЯ НЕЙРОМЕРЕЖЕВОЇ МОДЕЛІ ДЛЯ ІДЕНТИФІКАЦІЇ ДАНИХ ЗА РЕНТГЕНІВСЬКИМИ ЗНІМКАМИ

Оскільки рання діагностика може покращити результати лікування та збільшити час виживання пацієнтів з раком молочної залози, важливо розробити методи виявлення раку молочної залози. Конволюційна нейронна мережа (CNN) може автоматично отримувати функції з зображень, а потім проводити класифікацію. Однак для підготовки CNN з нуля потрібна велика кількість мічених зображень, що неможливо для деяких видів медичних зображень, таких як мамографічні зображення пухлин. Перспективним

рішенням є застосування трансферного навчання в CNN. Для початку потрібно протестувати три методи навчання на базі даних MIAS: 1) навчити CNN з нуля, 2) застосувати попередньо підготовлену модель VGG-16 для вилучення функцій з вхідних мамографів та використати ці функції для тренування нейронної мережі (NN) -класифікатор, 3) оновити ваги в декількох заключних шарах попередньо підготовленої моделі VGG-16 шляхом зворотного розповсюдження (тонкої настройки) для виявлення аномальних областей. Виявляється, що метод 2) є ідеальним для дослідження, оскільки точність класифікації тонкої настройки моделі була лише на 0,008 вище, ніж у моделі вилучення функцій, але часова вартість моделі вилучення функцій становила лише приблизно 5% від вартості тонкої настройки. Тоді ми використовуємо метод 2) для класифікації регіонів: доброякісних проти нормальних, злоякісних проти нормальних та аномальних проти нормальних з бази даних DDSM з 10-кратною перехресною валідацією. Середня точність перевірки збільшиться на рівні приблизно 0,905 для аномальних та звичайних випадків, і явного перевиконання не буде. Це дослідження показує, що застосування трансферного навчання в CNN дозволяє виявити рак молочної залози за допомогою мамографії, а навчання класифікатора NN шляхом вилучення ознак є більш швидким методом у трансферному навчанні. Наша мета - подолання проблеми класифікації бінарних зображень. Визначте наявність метастазів із цифрових гістопатологічних зображень 96 x 96px. Однією з ключових проблем є те, що метастази можуть бути такими ж маленькими, як одиничні клітини на великій ділянці тканини. Наш показник оцінки - площа під кривою робочої характеристики приймача (ROC).

Крива ROC - це графік істинної позитивної ставки проти хибної позитивної швидкості при різних порогах, а площа під кривою (AUC) дорівнює ймовірності того, що класифікатор класифікує випадково обраний позитивний екземпляр, вищий від випадково обраного від'ємного. Найкраще можливе рішення дасть AUC 1, що означає, що ми б класифікували всі позитивні зразки правильно, не отримуючи помилкових позитивних результатів.

Перш за все, нам потрібно проаналізувати наявні результати роботи з нашим набором даних за допомогою нейронних мереж. Як приклад, ми взяли дослідження Гарвардської медичної школи (BIDMC) та Массачусетського технологічного інституту (CSAIL), США та ExB Research and Development, Німеччина.

Гарвардська медична школа (BIDMC) та Массачусетський технологічний інститут (CSAIL), США вирішили проблему автоматичного виявлення метастазів раку, інтегруючи глибоке навчання та традиційні алгоритми машинного навчання. По-перше, вони тренували глибоку згорнуту

нейронну мережу (CNN) на $\sim 290K$ випадково вибраних пухлин і нормальних пластрів, які були вилучені із сотень тренуваних цілих слайдів. Архітектура їх CNN базувалася на 22-шаровому GoogLeNet, розробленому Szegedy та співавт. (CVPR 2014). Потім застосували підготовлену глибоку модель до частково перекриваючих патчів від кожного WSI для створення теплових карт прогнозування пухлини. Оцінивши точність теплових карт, вони витягли додаткові навчальні патчі з помилково позитивних областей і підготували остаточну модель для створення теплових карт прогнозування пухлини, використовуючи комбінований набір навчальних даних. Для завдання класифікації пухлин на основі слайдів витягнув набір геометричних ознак з кожної теплової карти імовірності пухлини у навчальному наборі та навчив випадкового лісового класифікатора для оцінки ймовірності того, що кожен слайд містив метастатичний рак. Потім вони застосовують ці моделі (CNN та випадковий класифікатор лісу) до тестових зображень, щоб забезпечити слайд-оцінку ймовірності ракових метастазів. Для завдання сегментації пухлинної області на основі ураження застосовують поріг 0,90 до теплових карт вірогідності пухлини та прогнозують місце розташування пухлини як центр кожної прогнозованої області пухлини.

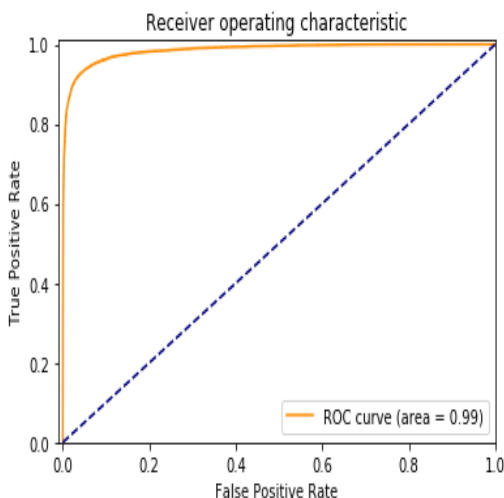


Рисунок 1 – ROC крива

ExB Research and Development, Німеччина, метод складається з чітко розробленої бінарної класифікації на основі глибоких залишкових мереж (ResNet). Вони розбивають кожне ціле слайдове зображення на плитки, що не

перекриваються, і присвоюють пухлину класу плитці, якщо плитка містить пікселі пухлини, а не пухлина в іншому випадку. Вони використовують цей параметр для тренування глибокої нейронної мережі на відповідне завдання бінарної класифікації. Зображення з набору даних Camelyon16 містять великі ділянки без тканин, які вкрай неефективні та вимагають багато часу для тренувань. Для їх видалення застосовується простий алгоритм попередньої обробки. Крім того, вони нормалізують наші зображення, щоб вони мали нульову середню і одиничну дисперсію, що було доведено для оптимізації процесу навчання. Архітектура нашої моделі складається з 34-шарового ResNet. Їхні експерименти показують, що ResNets постійно перевершував більш традиційні звивисті нейронні мережі при виконанні цього завдання з великими запасами точності класифікації 6-9%. Оцінки ROC та FROC також були значно гіршими. У контексті залишкових мереж вони експериментували на глибинах від 18 до 101 шарів. Результати показують, що мережі глибше 34 шарів мають тенденцію до зниження їх продуктивності. Потім вони збирають результати двох різних мереж, а також застосовують просту післяобробку для подальшого підвищення продуктивності свого алгоритму.

Зважаючи на наявні аналоги нейронних мереж для діагностики раку молочних залоз та мети даного процесу, а саме, найбільше наближення ROC кривої до 1, потрібно розробити свій алгоритм діагностики ракових пухлин.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Breast Cancer Detection Using Transfer Learning in Convolutional Neural Networks [Electronic resource]. – Access mode: <https://www.computer.org/csdl/proceedings-article/aipr/2017/08457948/13xI8A6FLUB>.
2. Receiver operating characteristic [Electronic resource]. – Access mode: https://en.wikipedia.org/wiki/Receiver_operating_characteristic
3. Deep Learning for Identifying Metastatic Breast Cancer / D. Wang, A. Khosla, R. Gargeya, H. Irshad, A. H. Beck. // arXiv:1606.05718, 1606.05718 (2016).
4. Ciresan D. Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. / D. Ciresan, A. Deep // In Pereira, F., Burges, C. J. C., Bottou, L. & Weinberger, K. Q. (eds) – Advances in Neural Information Processing Systems 25, 2843–2851 (Curran Associates, Inc., 2012).

МІКРОСЕРВІСНА АРХІТЕКТУРА ЯК СПОСІБ РОЗШИРЕННЯ ТА ВДОСКОНАЛЕННЯ МОНОЛІТНОГО ДОДАТКУ ДЛЯ ПИСЬМЕННИКІВ

Термін «Microservice Architecture» набув поширення в останні кілька років як опис способу дизайну додатків у вигляді набору незалежно розгорнутих сервісів. У той час як немає точного опису цього архітектурного стилю, існує якийсь загальний набір характеристик: організація сервісів навколо бізнес-потреб, автоматичне розгортання, перенос логіки від шини повідомлень до приймачів (endpoints) і децентралізований контроль над мовами і даними.

Використання мікросервісної архітектури надає розробнику наступні можливості щодо розширення проекту:

- модулі можна легко замінити в будь-який час: акцент на простоту, незалежність розгортання та оновлення кожного з мікросервісів;
- модулі організовані навколо функцій: мікросервіс по можливості виконує тільки одну досить елементарну функцію;
- модулі можуть бути реалізовані з використанням різних мов програмування, фреймворків, сполучного програмного забезпечення, виконуватися в різних середовищах розробки; виконуватися в різних середовищах контейнеризації, віртуалізації, під керуванням різних операційних систем на різних апаратних платформах
- архітектура симетрична, а не ієрархічна: залежності між мікросервісами однорангові.

Філософія мікросервісів фактично копіює філософію Unix, згідно з якою кожна програма повинна «робити щось одне, і робити це добре» і взаємодіяти з іншими програмами простими засобами.

Для організації мікросервісної архітектури на базі монолітної будуть використані наступні технології та програмне забезпечення, які використовуються для створення клієнт-серверної архітектури на стеку JavaScript технологій: серверна частина складається з Node.js, Koa, MongoDB, Redis; клієнтська була розроблена за допомогою таких технологій як React.js, React-Bootstrap, React Redux, React Router, React Router DOM. Для комунікації мікросервісних модулів було використано open source message broker NATS - система обміну повідомленнями з відкритим вихідним кодом. Сервер NATS написаний на мові програмування Go. На рис.1 представлена

робота та структура системи.

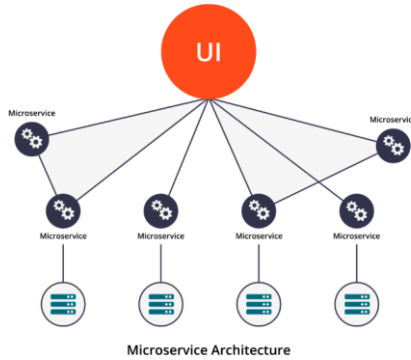


Рисунок 1 – Структура та функціонування мікросервісної системи

Представлена система для розширення монолітної архітектури є кросплатформенною та працює під керуванням операційних систем Windows та Unix/Linux. Робота системи на цих браузерах була перевірена практичним шляхом.

У порівнянні зі старою версією програми мікросервісне вдосконалення надає додатку більш короткий час запуску і можливість розгортання мікросервісів незалежно один від одного, що є дійсно вигідні для CI / CD. Мікросервіси не прив'язані до технології, яка використовується в інших сервісах. Це означає що є можливість використовувати кращі технології підгонки. Старі сервіси можуть бути швидко переписані для використання нових технологій. Саме можливість швидко адаптуватися до нових технологій може ще довго залишати різні проекти актуальними, та розширювати їх потенціал та можливості.

Таким чином була створена мікросервісна система для створення та читання художніх творів, яка складається із серверної та клієнтської частини. На відміну від старої версії додатку, нова характеризується більшою швидкістю обробки даних, має безліч варіантів вдосконалення та є простою в обслуговуванні

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ньюмен, С. Создание микросервисов [Текст] / Ньюмен С. – СПб. : БХВ-Петербург, 2018. – 304 с.
2. dev.to [Електронний ресурс]. – Режим доступу: <https://dev.to/t/webdev>

УДК 004.41

Тіменко А.В.¹, Дринов Д.О.².

¹ асист. НУ «Запорізька політехніка»

² студ. гр. КНТ-519м НУ «Запорізька політехніка»

CMS WORDPRESS ТА PHP FRAMEWORK LARAVEL, ТА СФЕРИ ЇХ ВИКОРИСТАННЯ

Wordpress - є світовим лідером серед CMS. Існує давно, підтримується творцями. Популярність, судячи з трендам Google, повільно знижується, але все одно залишається недосяжною для конкурентів.

Laravel - відносно молодий PHP-фреймворк, що стрімко набирає обороти. Швидко обігнав інші PHP-фреймворки, в 2018 р за популярністю серед розробників обігнав інші серверні фреймворки (не тільки PHP).

Ми будемо порівняти за такими критеріями:

- доступ до БД;
- структура коду;
- внутрішні залежності та зовнішні залежності;
- можливість адміністрування веб-ресурсу;
- авто-тестування;
- обробка помилок;
- універсальність.

CMS значно прискорює розробку простого шаблонного сайту. У сайту відразу готова админка і її не треба писати окремо, на відміну від розробки на фреймворку. Однак це швидкість створення сайту досягається за рахунок шаблонності, обмеженості або зайвої універсальності CMS.

При бажанні розширити функціонал, додати щось таке, можливо, доведеться прямо воювати з CMS або йти на якісь компроміси. У випадку з фреймворком, таких проблем немає. У фреймворк часто тільки самий необхідні функції і хороший, налагоджений, отрефакторений код. У CMS, особливо, якщо вона передбачає сторонні модулі, можна зустріти дике пекло в коді і навіть шкідливий код, який може, наприклад додавати посилання на ліві сайти в твій проект.

Ведучи розробку за допомогою фреймворка, ти сам будуєш взаємодія користувача з функціями і даними і сам вирішуєш як це буде виводитися в браузер. У CMS це все вирішили за тебе. Ймовірно, це і непогано, адже якщо у розробника недостатньо досвіду, то з фреймворком або чистою мовою можна повернути такого, що в підсумку CMS виявиться дуже хорошим варіантом.

Використовуючи CMS, ми прив'язуємося до структури, створеної розробниками. А Framework дає повну свободу дій. За нас написали основу,

фундамент, а далі бери і твори.

Зараз я б міг порадити WordPress для тематичних проєктів, коли потрібно про щось розповісти, наприклад, про товар, послугу, компанію, заході. У цій ситуації розробник виступає в ролі розробника кастомної теми. Більшу частину функціональності (роутинг, шаблонизатор, робота з БД і админка) WP дає з коробки. Краще використовувати:

- промо-сайти;
- корпоративні сайти;
- блоги.

Ієрархію CMS WordPress подано на рис. 1



Рисунок 1 – Ієрархія CMS WordPress

Фреймворк Laravel я б взяв для проєктів, де потрібно реалізувати якусь логіку. Частіше використовується для:

- B2B і B2C портали для клієнтів (корпоративні портали;
- HR-портали;
- Інтернет магазини;
- Аукціонні і тендерні майданчики;
- Системи онлайн-запису і бронювання.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Дронов В. Характеристики Laravel. Швидка розробка сучасних динамічних Web-сайтів на PHP, MySQL, HTML і CSS / В. Дронов. – БХВ-Петербург, 2016. – 768 с.

2. WordPress [Електронний ресурс]. – Режим доступу: <https://wordpress.org/>.

УДК 004.384

Тіменко А.В.¹, Опара О.С.²

¹ асист. НУ «Запорізька політехніка»

² студ. гр. КНТ-519м НУ «Запорізька політехніка»

НЕЙРОННІ МЕРЕЖІ У СУЧАСНОМУ СВІТІ

Прояв інтересу до штучних нейронних мереж було зумовлено роботами піонерів в цій справі - У. Маккалоха (W. McCulloch) і У. Піттса (W. Pitts). У 1943 році увагу громадськості привернула робота з назвою «Логічне числення ідей, що відносяться до нервової діяльності», в якій вони запропонували математичну модель нейрона і сформулювали принципи побудови штучних нейронних мереж, згідно з розробленою ними моделлю функціонування головного мозку. Багато вчених з ентузіазмом почали пропонувати свої рішення і нові архітектури ШНМ. Так, наприклад, Д. Хебб (D. Hebb) в 1949 році висловив ідеї про характер з'єднань нейронів мозку і їх взаємодії, представивши вперше методи навчання ШНМ, а великий нейрофізіолог Френк Розенблат (F. Rosenblatt) запропонував в 1957 році свою модель нейронної мережі - перцептрон, і архітектуру першої в світі моделі нейрокомп'ютера Mark.

У 1982 році американським біофізиком Дж. Хопфілдом була запропонована архітектура нейронної мережі, яка використовує енергетичний підхід, що моделює асоціативну пам'ять. Це був новий поштовх для розвитку цієї галузі, і в 1986 році Дж. Хінтон (J. Hinton) зі своїми колегами опублікував статтю з описом нової моделі нейронної мережі і алгоритмом її навчання[1].

На схематичному малюнку представлений біологічний нейрон (рис. 1).

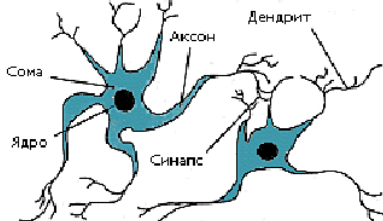


Рисунок 1 – Модель біологічного нейрона

Схема його дії досить проста. Нейрон отримує імпульси від аксонів інших нейронів через дендрити (приймачі) і передає сигнали, що згенеровані тілом клітки, вздовж свого аксона (передавача), який в кінці розгалужується на волокна. На закінченнях цих волокон знаходяться спеціальні утворення -

синапси, які впливають на величину імпульсів. Принцип функціонування нейронної мережі полягає у програмному або апаратному втіленні математичної моделі (рис. 2), що побудована за принципом біологічного нейрона, що був описаний раніше.

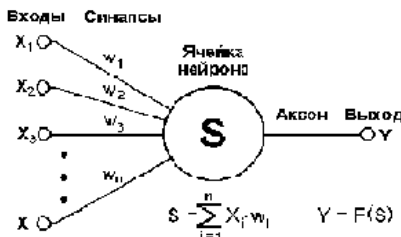


Рисунок 2 – Загальний вигляд штучного нейрону

Під навчанням ШНМ розуміється налаштування коефіцієнтів для ефективного виконання спеціалізованого завдання. Найчастіше пропонується навчальна вибірка (вхід-вихід), і за допомогою ітеративного наближення ШНМ налаштовує свої ваги необхідним чином, що покращує функціонування мережі.

У цьому і полягає перевага систем на основі ШНМ перед системами, в яких функціонування здійснюється за суворо визначеними правилами.

У спрощеному варіанті процес створення та проектування ШНМ можна уявити нижчеописаним способом. Визначається модель процесу, для якого створюється ШНМ, діючі фактори, доступна інформація. Потім, згідно з певних умов, визначаються правила, методи, модифікації вагових коефіцієнтів - з'являється алгоритм навчання. Така мережа вже здатна до навчання і існування.

Для навчання ШНМ використовуються три парадигми навчання: «з учителем», «без вчителя» і змішана.

У першому випадку на входи нейронної мережі подаються значення, необхідні результати яких відомі заздалегідь. Коефіцієнти входів модифікуються таким чином, щоб результатом роботи мережі були відомі заздалегідь результати. Модифікований варіант навчання з учителем передбачає, що відома тільки оцінка правильності виходу нейронної мережі, але не правильні значення виходу. У другому випадку проводиться структурування і розподіл даних за категоріями, згідно з певних зв'язків між входами. Змішаний варіант навчання передбачає під собою навчання деякої частини нейронів з учителем, а інша частина навчається іншим методом.

Можна виділити такі типові задачі для ШНМ:

- Класифікація образів. Завдання полягає у вказівці приналежності

вхідного образу, представленого вектором ознак, одному або декільком попередньо визначеним класам;

- Кластеризація / категоризація. При вирішенні задачі кластеризації, яка відома також як класифікація образів «без вчителя», відсутня навчальна вибірка з мітками класів.;

- Апроксимація функцій. Завдання апроксимації полягає в знаходженні оцінки невідомої функції;

- Передбачення / прогноз. Нехай задані n дискретних відліків $\{y(t_1), y(t_2) \dots, y(t_n)\}$ в послідовні моменти часу t_1, t_2, \dots, t_n . Завдання полягає в передбаченні значення $y(t_n + 1)$ в деякий майбутній момент часу $t_n + 1$;

- Оптимізація.

Можна виділити такі популярні архітектури ШНМ як багат шаровий перцептрон, рекурентний перцептрон, асоціативна пам'ять[2].

Багат шаровий перцептрон

Найвідоміша і дуже стара архітектура, в якій йдуть підряд кілька шарів нейронів - вхідний, один або кілька прихованих шарів, і вихідний шар. Майже завжди навчається методом зворотного поширення помилки - що автоматично означає, що ми повинні надати для навчання набір пар «вхідний вектор - правильний вихід». Тоді вхідний вектор відправиться на вхід мережі, послідовно будуть розраховані стану всіх проміжних нейронів, і на виході утворюється вихідний вектор, який порівнюється з правильним. Розбіжність дасть помилку, яку можна поширити назад зі зв'язків мережі, обчислити внесок в підсумкову помилку кожного нейрона, і скорегувати його ваги, щоб її виправити. Повторивши цю процедуру багато тисяч разів, можливо вийде навчити сет

Сильні сторони - вивчена з усіх боків, добре працює на своїх завданнях, якщо на деякій задачі не працює.

Слабкі сторони - невміння працювати з динамічними процесами, необхідність великої навчальної вибірки.

Рекурентний перцептрон

На перший погляд схожий на звичайний перцептрон, єдина істотна відмінність полягає в тому, що його виходи потрапляють йому ж на входи, і беруть участь в обробці вже наступного вхідного вектору. Тобто, в разі поворотного перцептрону має місце не набір окремих, нічим не пов'язаних образів, а певний процес, і значення мають не тільки самі входи, але і те, в якій послідовності вони надходять. Через це виникають відмінності в методи навчання - використовується те ж саме зворотне поширення помилки, але для того, щоб помилка потрапила по рекурентному зв'язку в минуле, використовуються різні хитрощі. В іншому ж ситуація схожа на звичайний перцептрон.

Сильні сторони: мережа дуже хороша для роботи з динамічними

процесами.

Слабкі сторони: якщо мережа не працює, зрозуміти помилку важко, адже у процесі навчання може перейти в режим самозбудження, а якщо рішення отримано – важко зрозуміти, чи можна досягти кращих результатів.

Асоціативна пам'ять

Це широкий клас мереж, які в тій чи іншій мірі нагадують архітектуру Хопфілда, яка складається з одного шару нейронів, виходи якого надходять на його входи в наступний момент часу. Цей шар служить і входом мережі і її виходом - значення на нейронах, що утворилися в кінці роботи, вважаються відповіддю мережі. Ця мережа змінює свої статки з плином часу до тих пір, доки стан не перестане змінюватися. Властивості вагової матриці обрані таким чином, щоб стійкий стан завжди гарантовано досягався (і зазвичай це відбувається за кілька кроків). Така мережа пам'ятає кілька векторів, і при подачі на вхід будь-якого вектору, може визначити, на який із запам'ятали він найбільше схожий. Двошарова модифікація цієї мережі може запам'ятовувати вектори не по-одному, а по парам різної розмірності.

Сильні сторони - дуже швидке навчання (тому що замість градієнтного спуску вирішується система рівнянь), можливість видалити образ з пам'яті або додати в пам'ять, не торкнувшись інші, деякі властивості такої пам'яті нагадують властивості мозку, і їх вивчення цікаво з такої позиції.

Слабкі сторони - дуже вузький клас вирішуваних завдань, невміння узагальнювати приклади, максимальний обсяг пам'яті жорстко пов'язаний з розмірністю вектору, що запам'ятовується.

Нейронні мережі активно використовуються у багатьох сферах сучасного світу, серед яких бізнес та економіка, медицина, інтернет, робототехніка, охоронні системи, обробка інформації та ін.

Розробка ефективних нейронних мереж ведеться вже більше 70 років, але до недавнього часу швидкість роботи нейронних мереж була занадто низькою, щоб вони могли отримати широке поширення, і тому такі системи в основному використовувалися в розробках, пов'язаних з комп'ютерним зором, а в інших областях застосовувалися інші алгоритми машинного навчання, але вже сьогодні велика продуктивність сучасних комп'ютерних систем та велика кількість доступних даних для навчання нейронних мереж зробила їх легкими та доступними у використанні, і популярність використання нейронних мереж тільки збільшується[3].

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Зачем нужны нейронные сети? [Електронний ресурс]. – Режим доступу: <http://md-it.ru/articles/html/article57.html>

2. Краткий обзор популярных нейронных сетей [Електронний ресурс]. – Режим доступу: <http://habr.com/ru/post/83781/>

УДК 004.42

Точилін С.Д.¹, Демченко М.М.², Точилін Д.С.³

¹канд. фіз.-мат. наук, доц. НУ «Запорізька політехніка»

²студ. гр. КНТ-519м НУ «Запорізька політехніка»

³інженер-програміст, КП НПК «Іскра»

КРОСПЛАТФОРМНИЙ ВІРТУАЛЬНИЙ НИЗЬКОЧАСТОТНИЙ ГЕНЕРАТОР

У наш час віртуальні прилади (ВП) знаходять широке застосування. Їх розробка є актуальною задачею.

У даній роботі розроблено кросплатформний віртуальний низькочастотний генератор з робочим діапазоном частот від 20 до 10^3 Гц. ВП складається зі звукової плати комп'ютера і програми ST_Sound.

Прикладна комп'ютерна програма ST_Sound створена за допомогою мови програмування JavaTM. Вона має графічний інтерфейс користувача, який дозволяє запускати та припиняти роботу генератора, а також установлювати частоту і амплітуду електричних синусоїдальних коливань, які генеруються ВП.

Вікно додатка ST_Sound при генерації кросплатформним віртуальним генератором коливань із частотою 3 кГц зображено на рис.1.

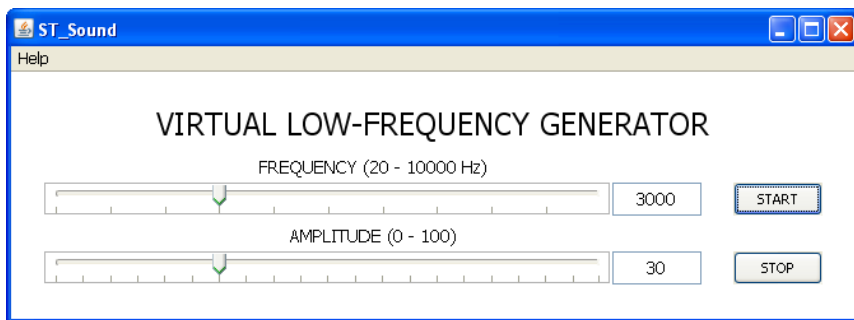


Рисунок 1 - Вікно програми ST_Sound при генерації ВП
низькочастотних коливань

Надалі передбачається модернізувати розроблений ВП. Зокрема, забезпечити можливість кросплатформному віртуальному генератору генерувати низькочастотні коливання прямокутної форми.

УДК 004.42

Точилін С.Д.¹, Нестюрич С.І.², Точилін Д.С.³

¹канд. фіз.-мат. наук, доц. НУ «Запорізька політехніка»

²студ. групи КНТ-519м НУ «Запорізька політехніка»

³інженер-програміст, КП НПК «Іскра»

КОМП'ЮТЕРНА ПРОГРАМА ДЛЯ ОЦІНКИ ВАРТОСТІ ОБ'ЄКТІВ НЕРУХОМОСТІ НА ОСНОВІ НЕЙРОННОЇ МЕРЕЖІ

У наш час нейромережеве програмування знаходить широке застосування в різних областях людської діяльності. При цьому розробка прикладних програм на основі нейронних мереж є актуальною задачею.

У даній роботі розроблена кросплатформна комп'ютерна програма ST_Realtor для оцінки вартості об'єктів нерухомості на основі нейронної мережі. Програма ST_Realtor створена за допомогою мови програмування JavaTM. Вона використовує алгоритм зворотного поширення помилки і має графічний інтерфейс користувача, який дозволяє:

1. Уводити та зберігати дані для аналізу, а також одержувати до них доступ.

2. Проводити навчання нейронної мережі та здійснювати оцінку вартості об'єктів нерухомості з її допомогою.

Для зберігання даних програма використовує СКБД PostgreSQL, яка вільно розповсюджується.

У той час нейронна мережа додатка ST_Realtor має три шари, константу навчання 0,1 та використовує для оцінки вартості об'єкта нерухомості п'ять вихідних параметрів.

Нами також проводилося зіставлення вартості окремих об'єктів нерухомості, що визначалися програмою ST_Realtor, з аналогічними, які були отримані для подібної нейронної мережі нейроемулятором Neural Network Wizard [1]. Як з'ясувалося, при роботі з тим же обсягом даних та пройшовши подібне навчання, ST_Realtor та Neural Network Wizard показували близьку за значенням вартість окремих об'єктів нерухомості.

Надалі передбачається модернізувати розроблений Java-додаток, зокрема, забезпечити можливість використовувати для оцінки вартості об'єктів нерухомості не менше десяти вихідних параметрів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Нейроемулятор Neural Network Wizard [Електронний ресурс]. – Режим доступу: <https://www.victoria.lviv.ua/library/students/nn/help/nn-wizard.pdf>

УДК 004.42

Точилін С.Д.¹, Демченко М.М.², Точилін Д.С.³

¹канд. фіз.-мат. наук, доц. НУ «Запорізька політехніка»

²студ. групи КНТ-519м НУ «Запорізька політехніка»

³інженер-програміст, КП НПК «Іскра»

КРОСПЛАТФОРМНИЙ ВІРТУАЛЬНИЙ ОСЦИЛОГРАФ

У даній роботі був розроблений кросплатформний віртуальний осцилограф (ВО) для дослідження електричних коливань із амплітудою від 0 до $\pm 2,5$ Вольт і частотою до 2 кГц. Він складався з плати Arduino Mega 2560, яка працювала спільно з повторювачем вхідної напруги на операційному підсилювачі ОРА340, та Java-програми ST_Oscillograph.

Вікно додатка ST_Oscillograph з осцилограмою синусоїдальних електричних коливань з частотою 200 Гц, яка була зареєстрована при роботі віртуального приладу, зображене на рисунку 1.

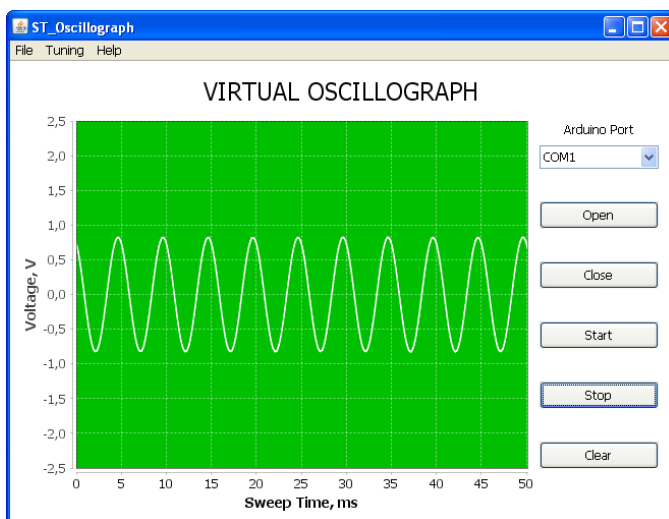


Рисунок 1 - Вікно програми ST_Oscillograph з зареєстрованою осцилограмою

На початку роботи з програмою за допомогою списку, що розкривається, та кнопки «Open» вибирався послідовний порт для роботи з платою Arduino Mega 2560 (для його закриття використовувалася кнопка «Close»). Запуск та зупинка роботи ВО здійснювалися при натисканні на

кнопки «Start» і «Stop», відповідно. Кнопка «Clear» очищала екран приладу від осцилограм, які були зареєстровані. Користувач програми за допомогою меню «File» мав можливість зберегти осцилограму у файлах формату *.csv та відкрити її для перегляду. Тим часом за допомогою меню «Tuning» при необхідності настраювалися область відображення (ОВ) осцилограми. Між тим інформація про створений ВО та особливостях його роботи була доступна при використанні меню «Help».

На рисунку 2 зображено вікно Java-програми ВО з осцилограмою коливань частотою 200 Гц, з іншою чим на рис. 1 областю відображення.

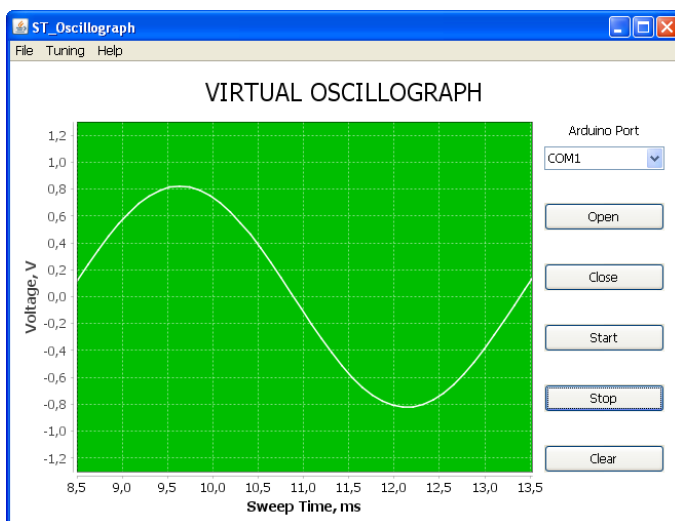


Рисунок 2 - Вікно програми ВО з іншою чим на рис. 1 ОВ осцилограми

Надалі передбачається модернізувати розроблений віртуальний осцилограф, зокрема, поліпшити його робочу напругу та дозвіл.

ВИКОРИСТАННЯ ТЕХНОЛОГІЇ RFID ДЛЯ СКЛАДСЬКОГО ОБЛІКУ

На сьогоднішній день нові комп'ютерні технології широко застосовуються в різних галузях промисловості для відстеження активів, в установках надання первинної медичної допомоги, в торгівлі (оптова та роздрібна), науково-дослідних лабораторіях, виробничих підприємствах, розподільних центрах, автопарках, ремонтно-будівельних управліннях та в інших місцях по всьому ланцюжку створення вартості. Однією з таких технологій є радіочастотна ідентифікація (RFID).

Системи з RFID (Radio-Frequency Identification) дозволяють ідентифікувати об'єкти без фізичного та візуального контакту. Ці системи містять зчитувач, або рідер, а також радіочастотну мітку, або транспондер. Рідер - це електронний пристрій для читання даних про об'єкт. RFID-мітка може зберігати дуже багато інформації, до 10000 байт можуть зберігатись на мікросхемі площею в 1 см² [1]. Переваги даних систем:

- безконтактний збір даних в режимі реального часу без візуального контакту;
- нечутливість до бруду та пошкоджень поверхні;
- більша відстань читання. RFID-мітка може зчитуватися на значно більшій відстані, ніж штрих-код;
- більший обсяг зберігання даних. RFID-мітка може зберігати значно більше інформації, ніж штрих-код;
- підтримка читання декількох міток;
- високий ступінь безпеки. Унікальне незмінне число-ідентифікатор, що привласнюється мітці при виробництві, гарантує високий ступінь захисту міток від підробки;
- можливість запису (перезапису) та зміни даних [2].

Саме тому метою даної роботи стало дослідження використання RFID для складського обліку. В даний час в більшості підприємств і, зокрема, складських господарствах, впроваджена система обліку продукції на основі штрих-кодів. RFID - це аналог штрих-коду, проте вона позбавлена недоліків систем на основі штрих-кодів, крім того, має ряд переваг, які перераховані вище.

Ідея полягає в тому, що товари маркуються спеціальними RFID-мітками, які потім при необхідності можна зчитати на відстані до 300 метрів RFID – зчитувачами.

RFID-мітка - мініатюрний чіп, який зберігає унікальний номер тега і інформацію і має можливість для передачі даних RFID-рідера. Як тільки RFID-мітка потрапляє в зону дії RFID-рідера, рідер фіксує факт передачі даних, зчитує інформацію з мітки і передає її в облікову систему, яка аналізує дані по заздалегідь заданим алгоритмам. Вся інформація пересилається в систему обліку даних і в будь-який момент може бути перевірена [3].

Отже, використання RFID-технологій на складах дозволить реалізувати:

- організацію ефективного обліку товару;
- контроль за переміщенням продукції складом;
- зменшення помилок персоналу при комплектації замовлень;
- скорочення кадрових та інших витрат на обслуговування складу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. RFID [Електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/RFID>
2. Технологія RFID [Електронний ресурс]. – Режим доступу: https://www.phoenixcontact.com/online/portal/ua?ldmy&urile=wcm:path:/uauk/web/main/products/technology_pages/subcategory_pages/Print_and_labelling_technique_RFID_technology/db66463f-f5ce-47e1-8a99-234b4a7120fd
3. Технологія RFID, метки, ридеры и ее применение [Електронний ресурс]. – Режим доступу: https://real-trac.com/ru/company/blog/princip_raboty_tehnologii_rfid_i_ee_primenenie/

УДК 004.896

Тягунова М.Ю.¹, Литвинов А.С.²

¹канд. техн. наук, доцент НУ «Запорізька політехніка»

²студ. гр. КНТ-519м НУ «Запорізька політехніка»

ВИКОРИСТАННЯ МОБІЛЬНОЇ ДОСЛІДНОЇ СТАНЦІЇ iCLebo KObuki У НАВЧАННІ

Останнім часом освіта у ВНЗ намагається не відставати від новітніх технологій та забезпечувати необхідною апаратною базою навчання студентів. Одним із сучасних напрямів розвитку на сьогодні є робототехнічні системи. iCLebo KObuki - це недорога мобільна дослідницька база, призначена для навчання і досліджень в галузі сучасної робототехніки. KObuki є джерелом живлення для зовнішнього комп'ютера, а також додаткових датчиків і виконавчих механізмів. Її високоточний одометр, з поправками відкалібрований на заводі, та гіроскоп, забезпечують точну навігацію [1].

KObuki спроектований для роботи від зовнішнього керуючого

комп'ютера і не має власного програмованого модуля. Для управління роботом виробник рекомендує використовувати нетбук на ОС Linux Ubuntu і фреймворк ROS (Robot Operating System). Обмін даними з роботом здійснюється через інтерфейс USB.

У середовищі ROS можна зчитувати інформацію з усіх датчиків робота і управляти його переміщенням відповідно до заданої програми [2].

Завдяки тому, що платформа Kobuki програмується, і для неї пишуться скрипти на мові програмування C ++, студент може отримати та вдосконалити навички програмування на цій мові, а також навчитись створювати алгоритми, що можна застосувати і в інших мовах програмування.

Також платформу Kobuki можна під'єднати до плати Arduino, завдяки чому можна отримати деякі навички у робототехніці.

Усі необхідні програмні комплекси та скрипти знаходяться у вільному доступі на сайті виробника, тому не потрібно затрачати матеріальні ресурси задля купівлі ПО, а також для навчання або ознайомлення з технологією не потрібно самому писати програми та скрипти для станції Kobuki.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. iCleo Kobuki [Електроний ресурс]. – Access mode: <https://dabit.industries/products/icleo-kobuki>

2. Знакомство с Kobuki [Електроний ресурс]. – Режим доступа: <https://iclebo.com.ru/blog/kobuki>

УДК 004.8

Тягунова М.Ю.¹, Циб О.О.²

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² студ. гр. КНТ-519м НУ «Запорізька політехніка»

ПРОГРАМНІ ЗАСОБИ ДЛЯ РОБОТИ З ПЛАТАМИ СІМЕЙСТВА ARDUINO

З кожним роком зростає популярність Інтернету речей, розумні пристрої займають все більшу частку нашого життя: збільшується кількість вбудованих систем, здешевлюється виробництво одноплатних комп'ютерів і плат з мікроконтролером. Все це робить їх доступними для широкої аудиторії, з'являються нові засоби автоматизації, збору і аналізу даних.

Плати сімейства Arduino - це електронні пристрої, що складаються з однієї друкованої плати, які здатні управляти різними датчиками, електродвигунами, освітленням, та здатні передавати і приймати дані [1].

Для реалізації роботи плат сімейства Arduino спочатку необхідно їх

запрограмувати. Програмна частина складається з програмної оболонки IDE. Найбільш популярними на даний момент є Arduino IDE, Programino, B4R, CodeBlocks for Arduino [2].

Метою роботи є аналіз існуючого IDE. Розглянемо їх можливості детальніше.

Arduino IDE - це відмінна платформа для моделювання на основі мікроконтролера плати Arduino UNO, що має багато варіацій, з купою проектів з відкритим кодом, підручників, форумів тощо для початківців, використовуючи простий код на основі C++.

Programino - середовище IDE для Arduino проста і зручна у використанні і повністю сумісна з оригінальним середовищем Arduino. Використовує оригінальні інструменти збірки від Arduino для компіляції і завантаження вашого коду за допомогою функції командного рядка `arduino`. Це IDE пропонує такий зручний спосіб швидкої розробки як автодоповнення коду. Тобто, не доведеться постійно дивитись в довідник з команд і методів Arduino.

B4R (Basic for Arduino) - це середовище розробки унікально тим, що використовує мову Basic, а не C++. Воно також підтримує функцію автодоповнення коду. В даному середовищі розробки можна ставити точки зупину, що корисно в процесі налагодження, а також використовувати закладки для більш швидкої навігації по коду.

CodeBlocks ID Arduino - це спеціалізований IDE з відкритим вихідним кодом, вдосконалений для розробки Arduino. Він уміщує в собі все, що знаходиться в сучасній IDE, включаючи створення папки коду, завершення коду, навігацію по коду, компіляцію, а також завантаження для Arduino. За допомогою спеціального майстра проектів легко створити готовий проект Arduino. Основна її відмінність - можливість писати код для мікроконтролерів і деякі інші платформ, не тільки для Arduino [3].

Існують такі випадки, коли необхідно реалізувати програмну частину в короткий термін. Тому не всі програмні оболонки підходять для даних випадків. Перевіривши та проаналізувавши існуючі програмні оболонки обрано програмне середовище Programino IDE, яке являється найбільш актуальним для реалізації проекту, що потребує швидкої розробки. Перевагами даного IDE є:

- підтримка наступних мов програмування C, C++, C Header, HTML & HTML 5, JavaScript, CSS;
- автодоповнення коду, часто використовувані команди можуть бути доступні через фрагменти та підказки;
- також присутній аналоговий плоттер - засіб для відображення показань у часі аналогових датчиків температури, вологості, тиску, освітленості та інших;

- огляду детальної інформації плат Arduino при програмуванні;
- функція експорту в Arduino IDE;
- функція імпорту бібліотек.

Таким чином, проведений аналіз показав, що якщо строки розробки системи з використанням плати сімейства Arduino дуже короткі, то найкращий вибір – це використання середовища Programino IDE.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Arduino платформа [Електронний ресурс]. – Режим доступу: <https://cxem.net/arduino/arduino49.php>.
2. Среды разработки (IDE) для arduino [Електронний ресурс]. – Режим доступу: <https://soltau.ru/index.php/themes/kompyutery-i-programmy/item/465-kakie-sushchestvuyut-sredy-razrabotki-ide-dlya-arduino>.
3. CodeBlocks Arduino IDE [Електронний ресурс]. – Режим доступу: <http://arduino4dev.com/codeblocks>.

УДК 004.896

Тягунова М.Ю.¹, Чумак І.Т.²

¹канд. техн. наук, доцент НУ «Запорізька політехніка»

²студ. гр. КНТ-519м НУ «Запорізька політехніка»

ДОСЛІДЖЕННЯ МОБІЛЬНОЇ РОБОТИЗОВАНОЇ ПЛАТФОРМИ iROBOT CREATE 2

У сучасному світі люди хочуть звільнити для своїх потреб більше часу. Одним з можливих варіантів є автоматизація повсякденних справ, наприклад, використання роботів або пристроїв, які характеризуються на прибиранні. Робот-пилосос - один з найпростіших варіантів, який оснащений штучним інтелектом, самостійно їздить, прибирає пил та бруд з підлоги.

На сучасному ринку доступна iRobot Create 2 Programmable Robot - мобільна роботизована платформа, розроблена для створення власних проектів на основі штучного інтелекту та з використанням зовнішніх інструментів, які встановлюються на робот, такі як камери, захоплення за допомогою вбудованого шаблону фронтальної панелі та додаткові інструменти, роздруковані на 3D принтері. iRobot Create 2 здатний працювати в ручному режимі під управлінням людини, в напівавтоматичному режимі (ухилення від перешкод і сходів) або зчитувати дані з сенсорів в повністю автоматичному режимі. Робот зберігає більшість можливостей пилососа, але в ньому немає щіток та лотка для збору сміття, тому його не можна використовувати, як пристрій для прибирання: він

призначений для розробки додатків [1]. Програмування робота виконується безпосередньо з комп'ютера через послідовний порт, або встановленого мікроконтролеру Arduino та міні-комп'ютеру Raspberry Pi. Робота с роботом не вимагає пайки і спеціалізованих кабелів та програмування на мовах низького рівня. Можливо створення багатьох проектів з використанням даної платформи.

У роботі розроблено систему пошуку предметів в кімнаті за допомогою вбудованої камери та штучного інтелекту. Для реалізації системи використовується нейрона мережа, яка ідентифікує предмети в кімнаті камерою та запам'ятовує місцезнаходження кожного з них. Людині з під'єданого до робота пристрою доступний перелік предметів, що ідентифіковані роботом та надається можливість отримати місцезнаходження на карті кімнати, потрібного предмету. Таким чином, розробка системи пошуку предметів в приміщенні дозволяє з мінімальним часом знайти потрібний предмет в кімнаті.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. ElectronicDesign [Електронний ресурс]. – Режим доступу: <https://www.electronicdesign.com/markets/robotics/article/21801421/running-the-robot-create-2>

УДК 004.051

Чубіч А.І.¹, Киричек Г.Г.²

¹ студ. гр. КНТ-526 НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

ВПРОВАДЖЕННЯ KANBAN-ДОШКИ ПРИ ОРГАНІЗАЦІЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Kanban - це метод поліпшення процесів розробки, що використовується гнучкими командами. Основною метою методу є отримання готового якісного продукту вчасно. Kanban починається з візуалізації, щоб процеси були на очах у команди. Для цього використовують дошки: фізичні та електронні. Дошка - це обов'язковий елемент для гнучкої методології. Кожен член команди отримує до неї доступ в будь-який час і бачить, на якому етапі перебуває завдання. Дошка дозволяє зробити робочий процес відкритим і зрозумілим для всіх фахівців, що важливо, коли у команди немає одного формального керівника [1]. Зазвичай команда складається з таких спеціалістів: власник продукту (Product Owner), програмісти і головний розробник, спеціалісти з тестування (QA) і головний QA (QA Lead), скрам-

майстер (Scrum Master) та бізнес-аналітик (Business Analyst) [2].

Після затвердження проектний рішень, прийнятих на етапі проектування, команда приступає до розробки. Весь функціонал розподіляється між програмістами, які розробляють алгоритми, пишуть вихідний код, виконують компіляцію і налагодження коду. У проекті є план процесу робіт. Спочатку ми його аналізуємо і поділяємо дошку на стовпці, які відображають етапи [3].

Виділимо наступні статуси для завдань: To do, In Progress, Code Review, Ready QA Testing, QA Under Test, Ready UAT Testing, UAT Under Test, UAT Done, Expected та Done (рис.1).

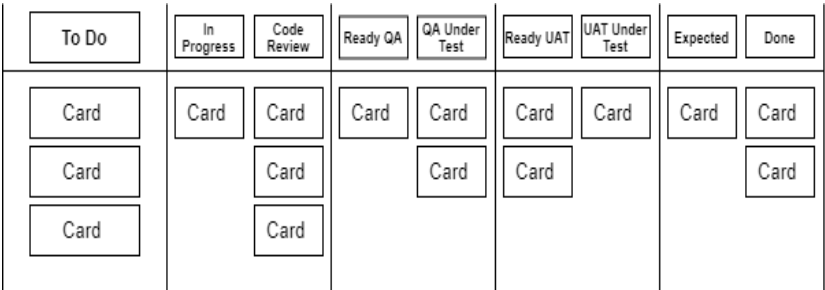


Рисунок 1 – Схема дошки за методологією Kanban

Виділимо 5 основних типів завдань, які будуть створюватись під час роботи над проектом: завдання (task), історія (story), епік (epic), поліпшення (improvement) та баг (bug). На початку циклу розробки створюється епік - це велике завдання, яке повністю описує окремий новий функціонал. Кожен епік складається з історій - описів вимог. Кожна історія є окремою вимогою, яка реалізується в системі. Історії можуть не входити до складу епіків, а будуть окремими самостійними об'єктами, які описують невеликий новий функціонал або зміни поточного. Крім того, створюються нові поліпшення (improvements) - це вимога до оновлення або невеликої зміни частини функціоналу. Покращення можуть додаватися вже після початку розробки, а також на етапі тестування. Тестувальники вносять свої пропозиції, які можуть спростити роботу з системою та зробити її більш комфортною для кінцевого користувача.

Для нових створених задач автоматично призначається статус To do. Цей статус означає, що задача створена, але поки не розпочата робота над нею. Розробник або тестувальник, на якого вона призначена, змінює статус на In Progress, коли приступає до виконання. Програміст, який розробляє або виправляє функціонал, описаний в завданні, вносить зміни на dev-оточенні.

Після цього статус завдання потрібно змінити на Code Review. Після встановлення цього статусу, на задачу автоматично призначається головний розробник, який перевіряє розроблений код. Якщо текст коду задовольняє вимогам, зміни додаються до тестового оточення. Задачі призначається статус Ready QA Testing [3].

Розробник може призначити задачу на тестувальника, який її створював або вже працював з нею. Також, можна залишити непризначеною (Unassigned) задачею і вільний тестувальник зможе почати роботу над нею. Коли починається етап тестування, QA-спеціаліст змінює статус на QA Under Test. Якщо під час перевірки, виникають питання або тестувальник знаходить баги, вони оформлюються в окремих задачах зі статусом To do. Коли тестування задачі закінчено, проблеми вирішені та функціонал відповідає поставленим до нього вимогам, тестувальник переміщає задачу у Ready UAT Testing статус.

Коли починається цикл UAT тестування, задачі переводяться спеціалістами у UAT Under Test статус та призначаються відповідні тестувальники. UAT тестування – це приймальне, призначене для користувача тестування. Будь-яка розробка або доопрацювання програмного забезпечення проходить завершальну стадію UAT-тестування. При цьому тестування проводиться бізнес-користувачами прийнятої системи. Але на практиці часто його проводять QA спеціалісти. Перевірки проводять на спеціальному оточенні, яке містить у собі всі зміни та доповнення, що відбувалися на етапах розробки та QA-циклу. Після успішного завершення перевірки, задачі переводяться до UAT Done.

Також передбачені статуси Expected та Done. As Designed призначається створеному багу, якщо він визнається не багом, а особливою можливістю (feature). Тобто, мається на увазі, поведінка системи, яка явно не відображена у вимогах, але є правильною з точки зору бізнес-логіки. Статус Done використовується для багів, які вирішено не виправляти. Окрім цього, цей статус виставляється для задач типу Task, коли вони виконані тестувальником або розробником. Цикл змін статусів відображено на рисунку 2.

Розробка за системою Kanban дає більше гнучкості, якщо під гнучкістю розуміти частоту зміни пріоритетів. На відміну від Scrum-розробки, де є чітке розподілення на спринти, під час яких потрібно виконати чітко запланований об'єм роботи, для Kanban вимоги до системи можуть змінюватись декілька разів на день, як і пріоритети задач. Окрім цього, не прийнято робити оцінку часу, який необхідний для виконання задачі. Це опціонально, команда вирішує сама, адже кожен з учасників самостійно організовує свою роботу.

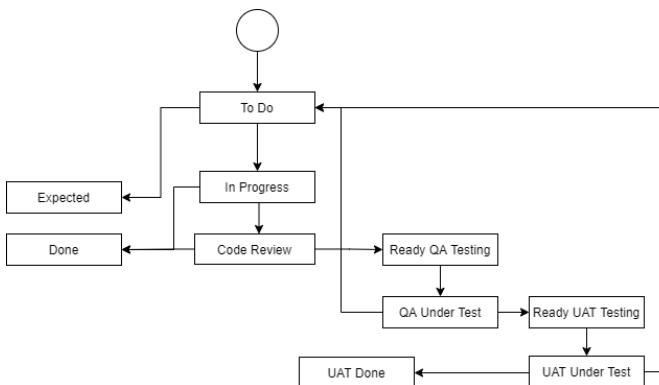


Рисунок 2 – Цикл змін статусів для задач

Висновки. Канбан-дошка та правила розроблено таким чином, щоб допомогти використовувати наявні ресурси найбільш ефективним способом, швидко знаходити проблеми і вирішувати їх в стислі терміни. Центральне місце в Kanban займає візуалізація процесу, яка допомагає відстежити стан задач, які виконуються на даний момент, хто над яким завданням працює і скільки часу потрібно для його завершення.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Стелман Е. Осягаючі Agile: Цінності, принципи, методології. / Е. Стелман, Д. Грін. – М.: Манн, Іванов і Фербер, 2019. – 448с.
2. Мартин Р. Быстрая разработка программ. Принципы, примеры, практика. / Р. Мартин, Дж. Ньюкирк, Р. Косс. – М.: Вильямс, 2004. – 752 с.
3. Кон М. Scrum: гибкая разработка ПО. / М. Кон. – М.: Вильямс, 2011. – 576 с.

УДК 004.05

Шавло Г.В.¹, Киричек Г.Г.²

¹ студ. гр. КНТ-516 НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

ВИКОРИСТАННЯ ВЕБ ТЕХНОЛОГІЙ В СИСТЕМАХ ВІРТУАЛЬНОЇ КОМЕРЦІЇ

Нині інтернет є високоінтелектуальною сферою, де відбувається обмін, зберігання і обробка великих масивів даних. Споживачами цих даних є

користувачі або кінцеві пристрої, що здатні, використовуючи потрібну інформацію, приймати відповідні рішення [1]. Реалізація віртуальних крамниць на даний час є одним з цікавих та перспективних інструментів онлайн-бізнесу.

Керуючи подібними системами є можливість реалізувати вимоги користувача, використовуючи правильний підхід до проектування та впровадження подібних систем та їх окремих модулів та інтерфейсів. При цьому інтерфейс визначає дані для зв'язку одного модуля системи з іншими [2].

Модуль може мати декілька інтерфейсів, які успадковуються у вигляді класів, використовуються в моделях системи і описуються будь-якою мовою програмування. Модулями можуть бути: програмні, сервісні, системні та службові; серверні, клієнтські і веб-серверні та веб-клієнтські; контейнери, патерни і екземпляри класів; а також системні (Big Data, Cloud Computing, Amazon та ін.). Кожен з них має специфікацію, вимог та правила їх взаємодії між собою [3].

Класифікація подібних систем та їх бізнес-платформ, надає змогу більш ретельно підійти до вибору веб технологій та мов програмування. Цю класифікацію можна представити трьома видами: крамниці спеціалізованих товарів (комп'ютери, комплектуючі, мережеві пристрої тощо); електронні супермаркети (все від техніки до одягу) та торговельні мережі віртуальних-крамниць (товари об'єднані у каталоги, з універсальною системою пошуку і замовлення товарів).

Для усіх віртуальних крамниць властивий обов'язковий набір елементів, таких як: каталог з розвинуеною структурою розділів; модуль реєстрації користувача; модуль виконання замовлення та модуль розрахунків в мережі, а також модуль виконання доставки кінцевому користувачу.

Етапи модульної реалізації системи: визначення структури і основних функцій системи; вибір архітектури, дизайну та стилістичної реалізації; проектування окремих модулів системи та реалізація її моделі; вибір веб технологій та мови програмування; реалізація клієнт-серверної архітектури; програмна та графічна реалізація окремих модулів системи та веб сайту в цілому; тестування системи за параметрами ефективності, завантаженості при використанні та дружності до користувача [4].

Веб технології є ефективним засобом створення Інтернет проектів різної складності та цільової спрямованості. Кожна з цих технологій має свої особливості, слабкі та сильні сторони. Наприклад, JavaScript є мовою яка дозволяє застосовувати складні речі на web сторінці кожного разу, коли на ній відбувається щось більше, ніж просто її статичне відображення [5]. Системи реалізовані на основі CGI, дозволяють ефективно розподіляти та організовувати різні дані, спрощуючи їх зберігання і структурування

інформації. CGI частіше всього використовується при створенні форумів, чатів, дошок оголошень і конференцій. Модель роботи подібної системи наведена на рисунку 1.

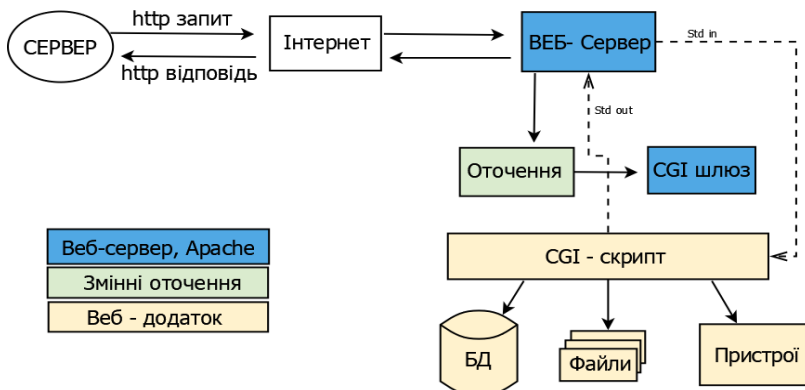


Рисунок 1 – Модель роботи системи

Особливості технології CSS в можливості здійснювати складне форматування сайтів. PHP є однією з найпоширеніших мов програмування, яка дозволяє додавати на сторінки інтерактивні елементи. Вона може використовуватися спільно з VBScript та ASP, що призначені для надання веб-сторінкам динамічності та інтерактивності.

Технологія DHTML дозволяє включати до складу сторінок такі інтерактивні елементи, як фон, що рухається, спливаючі меню або кнопки, які підсвічуються при наведенні курсору миші, а також рядок, що рухається і ін. При цьому вона має зрозумілу структуру.

Автором планується, при реалізації системи з підтримки віртуальної комерції, використовувати такі мови програмування як JavaScript і PHP, тому наводимо табличне порівняння їх особливостей та спільних функціональних можливостей (табл.1).

Висновки. Для розробки подібних віртуальних бізнес систем потрібно обирати технології, які ефективно реалізують функціональність усіх модулів системи та вирішують питання пов'язані з її використанням як користувачами так і клієнтськими пошуковими системами при аналізі усіх параметрів, вимог, повноти та різноманітності послуг. Веб-ресурс дозволить надати інформацію про компанію та її послуги стисло та інформативно. Окрім цього треба приділяти увагу відповідним стандартам, які забезпечать сумісність подібних систем і технологій. Це стандарти на використання і розміщення даних, обмін інформацією, використання метаданих і пошуку

інформації та забезпечення безпеки при виконанні розрахункових операцій.

Таблиця 1 – Порівняння функцій JavaScript і PHP

№	JavaScript та PHP (спільне)	JavaScript та PHP (різне)
1.	Для виконання завдань веб розробки, допомагають створювати привабливий інтерфейс з потужними внутрішніми функціями.	PHP для одночасної роботи с декількома запитами, використовує багатопоточність. У JavaScript для цього є цикл event loop в Node.js.
2.	PHP і JavaScript є мовами, що інтерпретуються. Код написаний на JavaScript, запускається у браузері, а PHP-код - на сервері	JavaScript наслідує концепцію ООП, має об'єкти і зв'язки між ними. У PHP, окрім об'єктно-орієнтованого підходу використовується і процедурний
3.	Обидві мають велику кодову базу для різних рішень, від систем управління контентом до потужних фреймворків.	JavaScript може бути вбудована в HTML, XML та AJAX. PHP тільки в HTML. Вона не може використовуватися з XML.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Nixon R. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 / R. Nixon. – " O'Reilly Media, Inc.", 2014. – 782 p.
2. Побединский Е. В. Проектирование веб-сайтов с использованием технологий PHP, HTML, CSS и WordPress / Е. В. Побединский. – Екатеринбург, 2018. – 115 с.
3. Лаврищева Е. Подход к моделированию систем и сайтов из готовых ресурсов / Е. Лаврищева, А. Рыжов. – Научный сервис в сети интернет, 2018. – С. 321-345.
4. Киричек Г.Г. Модель оцінки плагіату програмного коду на основі системи контролю версій / Г.Г. Киричек, О.О. Киричек // Восточно-Европейский журнал передовых технологий, 2012 – № 2/2. Вып. 56. – С. 25-28.
5. Заяц Т.А. Использование Web-технологий в создании Интернет-магазинов / Т.А. Заяц // Потребительская кооперация, 2011. – № 3. – С. 55–62.

ОНЛАЙН-СЕРВІС ВИВЧЕННЯ ІНОЗЕМНИХ МОВ

У час комп'ютеризації та автоматизації майже всіх сфер людського життя важливим критерієм є мінімум витрачених часу та ресурсів при максимальній ефективності.

При вивченні іноземної мови з репетитором (вчителем) основною проблемою є місце проведення та шлях до нього, і тому вибір вчителя та місця є нелегкою задачею. Саме тому виникає потреба в альтернативному рішенні, наприклад, перевести фізичне місце в віртуальне середовище – сайт, де відкривається багатофункціональний простір для дій.

Онлайн-сервіс – це альтернатива звичайному проведенню занять з викладачем, адже учень може не виходячи зі свого дому легко знайти необхідного йому викладача та провести урок з будь-якого місця [1]. Також можливо пройти курси, в яких буде інформація для вивчення та інтерактивна частина для практичного запам'ятовування [2].

Саме з цією метою було розроблено онлайн-сервіс, що дозволяє:

- надати користувачу інтуїтивно зрозумілий та зручний інтерфейс;
- стабільний зв'язок викладача та учня для проведення занять;
- можливість проведення групових занять;
- курси для самостійного опрацювання матеріалу;
- зручні способи оплати;
- стежити за стабільністю роботи сервісу

В результаті роботи розроблено систему з інтуїтивно зрозумілим і зручним інтерфейсом, яка надає користувачеві якісний зв'язок з викладачем. При цьому надано широкий вибір викладачів, щоб кожен учень міг знайти того, хто йому до вподоби.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Класифікація методів навчання [Електронний ресурс]. –Режим доступу: https://pidruchniki.com/14280824/pedagogika/klasifikatsiya_metodiv_navchannya.
2. Переваги і недоліки онлайн навчання [Електронний ресурс]. –Режим доступу: <https://openmind.com.ua/2018/01/09/skype-pros-cons/>

УДК 004.42

Щербак С.О.¹, Скрупський С.Ю.², Щербак Н.В.³

¹ студ. гр. КНТ-519м НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

³ старш. викл. НУ «Запорізька політехніка»

ПІДХІД ДО СПРОЩЕННЯ ПРОЦЕСУ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ІТ-індустрія одна з найбільших галузей в Україні та є ключовим компонентом майбутнього економічного успіху країни. У 2019 році українські компанії експортували за кордон ІТ-послуг на загальну суму 2,43 мільярда доларів [1].

ІТ-аутсорсингові компанії перед розробкою програмного забезпечення для замовника повинні узгодити умови так званого «залізного трикутника» проекту: функціональність, вартість, та час. Найбільшою проблемою є те, що замовник розраховує витрати менший бюджет, ніж той, який виявляється після узгодження необхідних функціональних та не функціональних вимог до програмного забезпечення. Тому у компанії розробника виникає необхідність спрощення процесу розробки програмного забезпечення, що дозволяє зменшити вартість.

Фактор, що найбільше впливає на вартість розробки програмного забезпечення – це кількість витрачених розробником годин. Найбільш трудомістким є створення back-end, а саме: створення бази даних, бізнес-логіки програмного забезпечення [2] та можливість редагування даних [3].

На сьогодні існує багато інструментів, які допомагають зменшити час роботи розробника на створення back-end:

- он-лайн конструктори сайтів [4];
- різноманітні DSL (Domain Specific Language) [5];
- інші програмні засоби;

Однак, існуючі інструменти найчастіше або дорого коштують, або є вузькоспеціалізованими й не дозволяють масштабувати в подальшому програмне забезпечення, що розроблюється. Тому вони не вирішують проблему зменшення вартості розробки програмного забезпечення.

В результаті виникає необхідність в розробці програмного забезпечення, яке б дозволило розробнику спростити процес створення back-end.

При цьому до програмного забезпечення висуваються наступні вимоги:

- можливість створювати ER-модель бази даних без написання коду;
- можливість редагувати дані бази даних без написання коду;
- створювати бізнес-логіку Web-додатку за допомогою DSL;

- легке та зручне налаштування програмного забезпечення на початку розробки Web-додатку.

Програмне забезпечення розроблено за допомогою наступних технологій: React, Node JS, PostgreSQL, GraphQL.

Для створення баз даних обрано систему PostgreSQL, запити до баз даних формуються за допомогою технології GraphQL [6]. Також технологій обумовлено тим, що на сьогоднішній день мова програмування JavaScript є найпопулярнішою та підтримує високий тренд розвитку[7]. JavaScript є основою технологій React та Node, які дозволять створювати інтерфейс користувача та бізнес-логіку. Технології PostgreSQL та GraphQL сумісні між собою та стеком технологій на базі мови JavaScript.

В результаті отримуємо програмне забезпечення, яке спрощує розробнику трудомісткий процес створення back-end, що дозволяє зменшити вартість розробки Web-додатку.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Мельник Г. Скільки Україні приніс експорт ІТ-послуг / Мельник Г. [Електронний ресурс]. – Режим доступу: <https://shotam.info/skil-ky-ukraini-prynis-eksport-it-poslugh/>
2. Вигерс Карл. Разработка требований к программному обеспечению / Карл Вигерс, Джой Бити. 3-е изд., дополненное – Пер. с англ. – М.: Издательство «Русская редакция»; СПб.: БХВ-Петербург, 2014. – 736 с.
3. Макконнелл Стив. Сколько стоит программный проект / Стив Макконнелл Питер, Пер. с англ. – М.: Издательство «Русская редакция»; СПб.: Питер, 2007. – 297 с.
4. Огляд 19 кращих конструкторів сайтів [Електронний ресурс]. – Режим доступу: <https://hostiq.ua/blog/site-builders/>
5. Введення в DSL. Частина 1 - Проблематика проектування і кодування [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/post/94259/>
6. Node.JS + React + GraphQL [Електронний ресурс]. – Режим доступу: <https://jetruby.com/ru/blog/node-js-react-graphql/>
7. Рейтинг мов програмування 2020 [Електронний ресурс]. – Режим доступу: <https://dou.ua/lenta/articles/language-rating-jan-2020/>

АНІМАЦІЯ РУХУ МОДЕЛІ СУПУТНИКА ІЗ QT QUICK 3D API

Космічна галузь має максимально ефективно використовувати переваги сучасних технологій, серед яких є і 3D-анімація, щоб задовольняти постійно зростаючим вимогам до якості програмного та апаратного забезпечення космічних апаратів (КА), а також спростувати процес їхньої експлуатації.

У рамках роботи над створенням моделюючого програмного забезпечення (ПЗ) симулятора для відпрацювання системи керування (СК) КА, окрім відображення числових результатів моделювання та їх інтерпретації у двовимірні графіки залежності, виникла необхідність у демонстрації руху 3D-моделі супутника відносно відповідних просторових осей (x , y , z).

Зважаючи на те, що в роботі вже використовувався фреймворк Qt, було розглянуто варіанти сумісних з Qt платформи-незалежних API для впровадження 3D-графіки – OpenGL та Qt Quick 3D.

OpenGL є низькорівневим процедурним API, який втілює імперативний підхід до створення графіки. З одного боку, це вимагає глибокого знання законів 3D-графіки і математичних моделей, з іншого боку – дає свободу впровадження різних інновацій. Проте розширення функціоналу стандарту OpenGL сторонніми бібліотеками полегшує і прискорює процес створення тривимірної сцени [1]. Наприклад, Qt Quick 3D, поєднуючи в собі QML / C++ API та маючи за основу OpenGL для рендерингу (візуалізації об'єктів), є модулем Qt для розробки програм з 3D-графікою.

Qt Quick 3D має низку переваг у порівнянні з «чистим» OpenGL: по-перше, це нативність даного API для проекту на Qt, по-друге, його актуальність й активне вдосконалення (останні зміни були внесені у 2019 р., для OpenGL – у 2017 р.), і, нарешті, простота у розумінні та використанні його головних властивостей – «Entity-Component-System» (систему складають сутності, які утворені з компонентів (примітивів) із властивостями) та «Task-Based Engine» (рушій Qt Quick 3D опитує кожен кадр на набір завдань для виконання разом із залежностями між ними) [2].

Основа для Qt Quick 3D становлять скрипти, написані декларативною мовою QML, які можуть містити вставки коду на JavaScript. Мова QML добре оптимізована, має метаоб'єктну модель і зручна у взаємодії з кодом на C++ (метаоб'єктний або об'єктно-орієнтований сигнально-слотовий зв'язок).

Перевагою QML також є те, що за замовчуванням вона є інтерпретованою, тобто нема потреби компілювати її в бінарний код для запуску, але це можна зробити з метою підвищення швидкості виконання коду, використавши Qt Quick Compiler [3].

Qt Quick 3D надає достатній набір визначених моделей даних і видів їх представлення. Так, наприклад, клас *QQuaternion* дає змогу створювати кватерніони для подання положення та куту повороту об'єкту в тривимірному просторі [4].

Щодо шляхів використання даного модуля, то можливо з початку розробки в IDE Qt Creator вибрати опцію «Qt Quick Application». Таким чином, весь проект буде заснований здебільше на QML, ніж на C++. Однак, такий спосіб підходить більше для створення мобільних та веб-додатків.

Для моделюючого ПЗ симулятора обраний інший спосіб впровадження технології Qt Quick, а саме – використання класу *QQuickWidget*, який базується на класі *QWidget*, тобто поводження з ним у коді таке ж зручне, як й з іншими віджетами. Однак, коли застосовується графіка із складною динамікою, такий підхід може передбачати зниження продуктивності ПЗ [5].

Для завантаження опису геометрії (mesh) 3D-моделей найчастіше використовують текстовий формат OBJ, тому ще одним плюсом на користь Qt Quick 3D є реалізація в ньому автоматизованого читання OBJ-файлів в той час, як OpenGL потребує власноручного написання коду для цього. Даний формат легко імпортується та експортується у всі популярні середовища створення 3D-графіки, серед яких було обрано «Blender» (вільно поширюване та невибагливе до системних ресурсів).

Отже, використання API високого рівня – Qt Quick 3D – дозволило за короткий термін доповнити ПЗ симулятора візуалізацією руху 3D-моделі супутника. У майбутньому підхід до опрацювання 3D-контенту може видозмінюватися і покращуватися у зв'язку з розвитком можливостей Qt.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Khronos OpenGL Registry [Електронний ресурс]. – Режим доступу: https://www.khronos.org/registry/OpenGL/index_gl.php
2. S. Harmer Overview of Qt3D 2.0 – Part 1. [Електронний ресурс] / S. Harmer – Access mode: <https://www.kdab.com/overview-qt3d-2-0-part-1>
3. Lee Zhi Eng Hands-On GUI Programming with C++ and Qt / Lee Zhi Eng – UK : Packt Publishing, 2018 – 235 p.
4. Офіційна сторінка Qt Documentation [Електронний ресурс] – Режим доступу: <https://doc.qt.io>
5. Шлее М. Qt 5.10. Профессиональное программирование на C++. / М. Шлее – СПб : БХВ-Петербург – 2018 – 328 с.

СЕКЦІЯ «ПРОГРАМНІ ЗАСОБИ»

УДК 004.93

Субботін С.О.¹, Литвин В.А.²

¹ зав. каф. НУ "Запорізька політехніка"

² асп. НУ "Запорізька політехніка"

АНАЛІЗ МЕТОДІВ ЦИФРОВОЇ ОБРОБКИ СИГНАЛІВ ДЛЯ ПОБУДОВИ ДІАГНОСТИЧНИХ МОДЕЛЕЙ

При побудові діагностичних моделей на основі ознак, що є розподіленими у часі відліками сигналів, виникає потреба скорочення розмірності даних. Це обумовлюється тим, що індивідуально відліки сигналу є малоінформативними як ознаки. Використання відомих методів відбору ознак [1] також є неефективним, бо вимагатиме великих витрат часу на перебір величезної кількості комбінацій первинних ознак, та може призводити до втрати частки інформації. Методи скорочення розмірності даних [2] можуть призводити до втрати фізичного смислу первинних ознак і не дозволяють отримувати інтерпретабельні моделі.

На відміну від розглянутих методів методи цифрової обробки сигналів (ЦОС) [3–5] дозволяють зменшити кількість ознак, але зберегти фізичний смисл, а також у більшості випадків підвищити рівень узагальнення даних та зменшити вплив шумів на якість моделей.

Мета роботи полягала у виборі методів цифрової обробки розподілених у часі сигналів для виділення діагностичних ознак.

Найбільш широко застосовним на практиці класом методів ЦОС є фільтрація [3]. Цифрова фільтрація передбачає перетворення вхідного сигналу у вихідний, використовуючи комбінацію певних перетворень. До цього класу належать ковзне середнє, мінімум, максимум, сума, добуток, які визначаються шляхом проходження вікна по вхідному сигналу і визначенні відповідної функції для набору послідовно розташованих відліків сигналу, що розглядаються у вікні. Перевагою таких методів є те, що вони нівелюють вплив викидів, але їхнім недоліком є те, що вони не дозволяють виявити складові сигналу.

Методи ЦОС у частотному просторі (спектральний аналіз) [4] призначені для виявлення наявності певних частот у сигналі. Вони представлені перетворенням Фур'є, яке перетворює інформацію про час або простір у компоненти магнітуди та фази для кожної частоти. Перетворення Фур'є є широко застосовним для обробки сигналів, що складаються з періодичних компонентів. Перевагою перетворення Фур'є є його універсальність, що обумовлена інтегровальними властивостями за усією

часовою віссю, що є добрим для стаціонарних сигналів. Проте це є недоречними для аналізу нестаціонарних сигналів, які складаються з неперіодичних компонентів.

Методи вейвлет-аналізу [6] використовують інтегральне перетворення, що являє собою згортку вейвлет-функції з сигналом. Це перетворення переводить сигнал з часового у частотно-часове подання. Перевагами вейвлет-перетворень є те, що вони мають ті ж переваги, що й перетворення Фур'є, але на додаток також можуть бути добре локалізовані як за частотою, так і за часом. Базисні вейвлети можуть реалізовуватися функціями різної гладкості. Недоліком вейвлет-перетворення є його складність.

З наведеного вище огляду випливає, що жодна з груп методів ЦОС не є ефективною застосовною для виділення діагностичних ознак у всіх можливих випадках. Отже, на практиці доречно перед побудовою діагностичних моделей вхідний сигнал послідовно перетворювати на основі різних перетворень у порядку зростання їхньої складності і намагатися будувати модель на основі результатів отриманих перетворень, мінімізуючи тим самим складність перетворень та обсяг множини штучних діагностичних ознак.

Після того, як буде отримана модель, що має прийнятний показник якості, набір її ознак варто піддати скороченню шляхом застосування методів відбору ознак [1]. Після чого модель варто побудувати для скороченого набору ознак, отриманих відбором перетворень методами ЦОС.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Биргер, И. А. Техническая диагностика / И. А. Биргер. – М.: Машиностроение, 1978. – 240 с.
2. Fodor, I. A survey of dimension reduction techniques. National Technical Report UCRL-ID-148494 [Electronic resource] / I. Fodor. – Livermore: Lawrence Livermore National Laboratory, 2002. – Access mode: <http://citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.8.5098>.
3. Смит, С. Цифровая обработка сигналов. Практическое руководство для инженеров и научных работников / С. Смит. – М.: Додэка XXI, 2008. – 720 с.
4. Van De Vegte, J. Fundamentals of Digital Signal Processing / J. Van De Vegte. – London: Prentice Hall, 2001. – 810 p.
5. Лайонс, Р. Цифровая обработка сигналов / Р. Лайонс. – М.: Бином-Пресс, 2006. – 656 с.
6. Малла С. Вэйвлеты в обработке сигналов. – М.: Мир, 2005. – 672 с.

ПОСТРОЕНИЕ ДИАГНОСТИЧЕСКИХ МОДЕЛЕЙ С ПРЕДВАРИТЕЛЬНОЙ ЦИФРОВОЙ ОБРАБОТКОЙ СИГНАЛОВ

Для своевременного обнаружения ненадежных изделий необходимо обладать диагностической моделью. Широко используемым инструментарием для решения подобных задач являются методы вычислительного интеллекта: нейронные [1] и нейро-нечеткие [2] сети, иммунные системы [3], деревья решений [4], общим достоинством которых является возможность извлечения знаний из данных.

Однако их эффективность применения существенно снижается, если данные являются многомерными, а признаки - индивидуально малоинформативными. В подобных ситуациях модели оказываются чрезвычайно избыточными, неинтерпретабельными и не обеспечивают обобщения данных.

Целью данной работы являлось создание метода построения диагностических моделей, свободного от перечисленных выше недостатков.

Для этого предлагается исходный набор N индивидуально малоинформативных признаков разбить на группы, объединив в каждой группе связанные между собой признаки (по заданному критерию), для каждой группы признаков определить интегральные характеристики как искусственные признаки, заменяющие первичные. При этом целесообразно использовать такие способы расчета искусственных признаков, которые будут сохранять физическую интерпретацию относительно первичных признаков.

Затем следует оценить индивидуальную информативность искусственных признаков и отобрать такое их подмножество, которое будет достаточно для построения диагностической модели. При этом рационально процесс отбора признаков сочетать с процессом построения модели, получив не только набор искусственных признаков, но и диагностическую модель, построенную на его основе.

Для определения критерия группирования исходных признаков следует учитывать природу задачи. Так, если исходные признаки представляют собой отсчеты сигнала (либо какого-нибудь его параметра), упорядоченные во времени, то следует группировать отсчеты по близости во времени.

Если же рассматривать более общий случай, когда признаки имеют

разную природу, то их целесообразно группировать по степени взаимосвязанности. Если несколько признаков являются взаимосвязанными, то можно предположить, что они, либо являются проявлениями какого-то скрытого фактора, либо дублируются между собой. В этом случае группирование таких признаков позволит выделить подобные признаки. Каждую группу взаимосвязанных признаков следует заменить на искусственные признаки, обобщающие исходные признаки и выделяющие ключевые характеристики признаков группы.

Метод, реализующий приведенные идеи, можно представить следующим образом.

Этап инициализации. Задать исходную выборку прецедентов. Занести в текущий набор признаков весь исходный набор признаков. Задать показатель качества модели Q и его приемлемое значение.

Этап анализа взаимосвязи признаков. Определить значения показателя взаимосвязи каждого i -го и j -го входных признаков $I_{i,j}$ одним из способов, описанным в [5–7]. Определить значения показателя взаимосвязи i -го входного и выходного признаков I_i , используя один из способов [5–7]. На основе характеристик признаков определить показатели их связности: $\tilde{I}_{i,j} = I_{i,j} e^{-\alpha(i-j)^2} e^{-\beta(I_i - I_j)^2}$, где α и β – параметры, регулирующие ширину окна для формирования группы признаков. После чего рассчитать значения показателя группирования: $\hat{I}_{i,j} = \{1 | \tilde{I}_{i,j} \geq \tilde{I}_{i,p}, p \neq i, p \neq j, p = 1, 2, \dots, N\}$.

Этап группирования признаков. Пока текущий набор признаков непустой, выполнять: сформировать новую группу признаков, определить входной признак индивидуально наиболее информативный по отношению к выходному признаку среди признаков в текущем наборе, занести его в новую группу признаков как центр группы и удалить из текущего набора; среди всех оставшихся признаков текущего набора выделить те, которые сильнее связаны с признаком-центром группы, чем с любым другим признаком в текущем наборе (т.е. те, для которых $\hat{I}_{i,j} = 1$, занести их в группу и удалить из исходного набора).

Этап конструирования искусственных признаков для групп. Для каждой группы исходных признаков определить значения интегральных характеристик на основе методов цифровой обработки сигналов [8–10].

Этап анализа искусственных признаков. Для набора искусственных признаков определить показатели, характеризующие их взаимосвязи с выходным признаком J_i . На основе текущего набора искусственных признаков построить модель и оценить ее показатель качества Q .

Этап построения модели и отбора признаков. Если значение Q оказалось неприемлемым, то добавлять в текущий набор признаков

поочередно признаки из исходного набора с наиболее высокой информативностью I_i до тех пор, пока остались недобавленные исходные признаки и не достигнуто приемлемое значение показателя качества модели. Как альтернативу или дополнение к добавлению к текущему набору искусственных признаков признаков исходного набора представляется также возможным определить искусственные признаки для всего исходного сигнала на основе методов ЦОС и добавить их к набору признаков.

Для ускорения процесса добавления признаков возможно добавлять признаки группами: вместо одного наиболее информативного исходного признака добавлять все исходные признаки, входящие в его группу.

Если значение Q для полученной модели оказалась приемлемым, то решить задачу отбора подмножества наиболее информативных признаков из входящих в текущем наборе признаков модели, посредством исключения индивидуально малоинформативных признаков до тех пор, пока значение Q для модели, синтезированной на основе набора признаков, будет приемлемым.

Для ускорения процесса отбора признаков возможно удалять исходные признаки группами: вместо одного наименее информативного исходного признака исключать все исходные признаки, входящие в его группу.

В конечном итоге мы получим модель, построенную на основе минимально достаточного числа первичных и искусственных признаков, которая будет обладать приемлемым из достигнутых значений показателя качества.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – СПб: Вильямс, 2005. – 1104 с.
2. Computational intelligence in fault diagnosis / eds.: V. Palade, C. D. Bocaniala, L. Jain. – London: Springer, 2006. – 362 p.
3. Dasgupta, D. Immunological Computation: Theory and Applications / D. Dasgupta, F. Nino. – Boca Raton: CRC Press, 2008. – 296 p.
4. Utgoff, P. E. Incremental induction of decision trees / P. E. Utgoff. – Machine learning. – 1989. – Vol. 2, issue 4. – P. 161-186.
5. Олійник, А. О. Інтелектуальний аналіз даних : навчальний посібник / А. О. Олійник, С. О. Субботін, О. О. Олійник. – Запоріжжя : ЗНТУ, 2011. – 271 с.
6. Биргер, И. А. Техническая диагностика / И. А. Биргер. – М.: Машиностроение, 1978. – 240 с.
7. Guyon, I. An introduction to variable and feature selection / I. Guyon, A. Elisseeff // Journal of machine learning research. – 2003. – № 3. – P. 1157–1182.
8. Смит, С. Цифровая обработка сигналов. Практическое руководство для инженеров и научных работников / С. Смит. – М.: Додэка XXI, 2008. –

720 с.

9. Van De Vegte, J. Fundamentals of Digital Signal Processing / J. Van De Vegte. – London: Prentice Hall, 2001. – 810 p.

10. Лайонс, Р. Цифровая обработка сигналов / Р. Лайонс. – М.: Бином-Пресс, 2006. – 656 с.

УДК 004.55

Соколянський О.В.¹, Пархоменко А.В.²

¹ асп. НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

ДОСЛІДЖЕННЯ ТА ВИКОРИСТАННЯ СУЧАСНИХ ФРЕЙМВОРКІВ ДЛЯ МОДЕРНІЗАЦІЇ ВІДДАЛЕНОЇ ЛАБОРАТОРІЇ

На даний момент PHP – одна з найпопулярніших скриптових мов програмування, за допомогою якої створюється велика кількість різних веб-проектів. Як відомо, технології повторного використання, що швидко розвиваються, дозволяють спростити процес реалізації програмних проектів [1]. Зокрема, PHP фреймворки дозволяють прискорити процес розробки веб-додатків; писати простий і якісний код; повторно використовувати код в проєктах; легко масштабувати проєкти; використовувати сучасні практики програмування; ефективніше тестувати програмний код та ін.

Для розробки веб-інтерфейсу віддаленої лабораторії RELDES раніше використовувався застарілий на сьогоднішній день фреймворк CodeIgniter. З метою удосконалення проєкта, зокрема зменшення часу адаптації нових членів команди розробників, а також використання додаткових можливостей сучасних фреймворків було вирішено перенести розробку на інший фреймворк. Дослідження сучасних фреймворків Laravel, Yii, Symfony та Zend дозволило обрати Yii в якості інструментарію RAID розробки [2]. В процесі модернізації проєкта, завдяки гарній структурі CodeIgniter, було успішно перенесено наступні модулі: Контролери, Моделі, Сервіси, Міграції.

Використання фреймворку Yii надає безліч нових можливостей, зокрема механізм для побудови реляційної обробки запитів бази даних; вбудовані віджети для відображення даних; вбудовані механізми для аутентифікації, авторизації, реєстрації користувачів. Це дозволить розширити функціональність віддаленої лабораторії та забезпечити ефективну взаємодію з користувачами.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Parkhomenko, A. Implementation of reusable solutions for remote

laboratory development /A. Parkhomenko, O. Gladkova, A. Sokolyanskii, V. Shepelenko, Y. Zalyubovskiy// iJOE, vol. 12 (07), 2016. – pp.24-29.

2. Обзор РНР фреймворков [Електронний ресурс] – Режим доступу: <http://unetway.com/blog/php-framework-review/>

УДК 004.5

Туленков А.В.¹, Пархоменко А.В.²

¹ асп. НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

ЗАБЕЗПЕЧЕННЯ НОРМАЛЬНОГО ТЕПЛОВОГО РЕЖИМУ РОБОТИ ЕЛЕКТРОЩИТОВОГО ОБЛАДНАННЯ СИСТЕМИ ДОМАШНЬОЇ АВТОМАТИЗАЦІЇ

Як показав аналіз особливостей процесу автоматизованого проектування електрощитового обладнання системи домашньої автоматизації від компанії Legrand, аналіз теплового режиму є важливою задачею, що потребує використання спеціального програмного забезпечення та методики формування рекомендацій фахівцям для попередження позаштатних ситуацій, які можуть виникнути при експлуатації обладнання (коротке замкнення, пожежа, тощо).

Програмне забезпечення XLPro3 від компанії Legrand дозволяє проводити розрахунок теплового балансу в електрощиті. Алгоритми цього розрахунку є закритими, але їх ефективність та точність перевірені численними лабораторними випробуваннями. При роботі в програмі, спочатку необхідно вибрати обладнання, провести проектування однолінійної схеми та виконати підбір необхідної шафи з каталогу. Програма враховує тип щита (серію та модульність шафи), контактні поверхні щита, ступінь захисту, коефіцієнт втрат кабелю. Для дослідження теплових характеристик щита необхідно вказати дані щодо температури в робочому приміщенні, вологості, запиленості. Крім того, задаються: тип навантаження, тривалість включення, коефіцієнт комутації та інші. Як результат, програма визначає величину розсіяваної потужності, діапазон робочих температур виробу та відповідність номінального струму та температури існуючим вимогам.

Тим не менш, проведені теоретичні розрахунки не гарантують надійної роботи системи, оскільки параметри обладнання можуть змінюватися в процесі експлуатації або бути порушені користувачем системи після підключення.

Тому, задача забезпечення нормального теплового режиму всередині

розподільного щита при його експлуатації є актуальною та потребує розробки спеціального програмно-апаратного комплексу для моніторингу температурних показників електрощитового обладнання системи домашньої автоматизації. Пропонується реалізувати систему сповіщень користувача засобами мобільного зв'язку про виникнення позаштатних ситуацій внаслідок порушень теплового режиму роботи обладнання.

УДК 004.93

Федорончак Т.В.

канд. техн. наук, доц. НУ «Запорізька політехніка»

МЕТОДИ ОЦІНЮВАННЯ СКЛАДНОСТІ ЗАДАЧ В AGILE МЕТОДОЛОГІЯХ

Для складання календарних графіків та розрахунку бюджету проекту з розробки програмного забезпечення широко використовується оцінювання задач в людино-годинах. Такий підхід до оцінювання зрозумілий для команд і простий в використанні, однак вимагає наявності чітких та детальних вимог до продукту, є досить трудомістким та часто може давати велику похибку.

Зазвичай розробникам досить важко з високою точністю визначити кількість годин, необхідних для виконання певних завдань. Однак порівняння декількох об'єктів між собою за розміром для людини є досить природним. Використовуючи цю особливість, в сучасних agile методологіях для оцінювань користувацьких історій використовуються відносні одиниці. Такий підхід направлений на скорочення витрат часу на оцінювання.

В agile оцінюється розмір задачі, беручи до уваги обсяг роботи, що необхідно виконати, її складність, ризики та невизначеність при виконанні. Методи оцінювання в agile є груповими, в обговореннях бере участь вся команда, беручи на себе зобов'язання. Розмір задачі не прив'язується до будь-яких фізичних одиниць і лише дозволяє порівняти розмір задач між собою.

Розглянемо найбільш поширені з методів оцінювання, що використовуються agile командами.

Метод розміру футболка використовується для швидкого оцінювання великих задач, епіків та тем. Кожній задачі за результатами обговорення командою ставиться у відповідність один із розмірів XS, S, M, L, XL.

Метод системи відер також використовується для швидкого оцінювання. На початку визначається система відер різного розміру, кожний з яких визначає деякий проміжок часу. Далі команда визначає, в яке мінімальне відро поміститься кожна задача.

Метод покеру планування. Цей метод є одним з найбільш популярних

але трудомістких. Для оцінювання задач використовуються пункти, значення яких входять в дещо змінений ряд Фібоначчі (0, 0.5, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89) або рідше є ступенем двійки. Спочатку команда обирає одну задачу, що є простою та всім зрозумілою, та оцінює її в 1 пункт. Далі методом голосування команда визначає яке з чисел поставити у відповідність кожній задачі у порівнянні із іншими вже оціненими задачами. З плином часу після декількох ітерацій розробки команда може визначити свою швидкість і прогнозувати строки завершення задач, що залишилися.

Метод голосування за точками. Цей метод передбачає використання спеціальних точок, що показують оцінку певного завдання кожним учасником. Кожний учасник команди отримує однакову кількість точок і розподіляє їх на свій розсуд для оцінювання складності задач. По закінченню голосування кожна задача має свою оцінку, що є сумою точок від всіх учасників.

Метод розділення до максимального розміру чи менше. В цьому методі команда визначає максимальний можливий розмір для задачі (наприклад, 1 людино-день). Далі при обговоренні задач команда визначає, чи не перевищує розмір поточної задачі максимально можливий, і якщо так, задача розбивається на підзадачі, розмір яких задовольняє обмеженню на розмір.

Метод упорядкування розмірів. Даний метод являє собою покрокову гру, мета якої впорядкувати всі задачі на єдиній шкалі від малого до великого розміру. Спочатку історії випадковим чином розміщуються на шкалі. Далі кожен учасник команди по черзі або зміщує картку з задачею на шкалі на одне ділення, або ініціює обговорення історії з командою, або пропускає свій хід. Гра закінчується, коли всі учасники пропускають ходи, тобто всі задачі упорядковані.

Як можна побачити, різноманіття використовуваних методів оцінювання є дуже великим і не існує єдиного сталого підходу. Кожна agile команда на свій розсуд визначає правила оцінювання задач, підлаштовуючи існуючі методи під себе.

УДК 65.012.61

Колпакова Т.О.¹

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

ОСОБЛИВОСТІ ПІДБОРУ КОМАНДИ ДЛЯ РОЗРОБКИ WEB-ПРОЕКТІВ ЗА МЕТОДОЛОГІЄЮ AGILE

В світовій практиці існують різні підходи до планування розробки проектів в сфері інформаційних технологій. В роботі було проведено

порівняння класичних методологій, як то Waterfall, та нових "гнучких" методологій на основі Agile.

Було виявлено, що web-проектам притаманні певні специфічні властивості, а саме мінливість вимог, постійне розширення та доопрацювання переліку функцій, розмиті межі спеціалізації та професійних навичок учасників проекту.

Ці властивості в достатній мірі покриваються принципами сімейства методологій Agile, ідея якого полягає в тому, що увагу варто приділяти насамперед комунікаціям всередині команди розробників та не встановлювати жорсткі рамки виконання, що дозволяє вносити зміни в створюване програмне забезпечення навіть на пізніх стадіях реалізації без втрати якості продукту [1]. В рамках сімейства методологій Agile найбільш популярними є два підходи: Scrum та Kanban.

Scrum має чіткий розклад виконання задач і передбачає постійну комунікацію між командою та замовниками, тож є більш ефективним для продуктових команд, тобто команд, що працюють на етапі розробки та постачання програмних продуктів [2-3].

Kanban є менш стандартизованим в контексті розкладу і є більш ефективним на пізніх етапах розробки, наприклад тестування або підтримки після релізу. Він також є ефективним в стартапах, які не мають чіткого плану, але активно працюють над розробкою, змінюючи вимоги "на льоту" [4].

В роботі було описано ролі учасників, яких необхідно залучати до розробки web-проекту, та виявлено перелік вимог, що ставляться до розробників при використанні методології Scrum.

Були висвітлені основні проблеми, що виникають при формуванні команди розробки. При залученні недостатньої кількості учасників проект буде розроблятися занадто довго, і деякі вимоги будуть застарівати швидше, ніж відбудеться їхня реалізація. При залученні зовеликої кількості учасників більше часу буде витратитись на узгодження задач та вирішення конфліктів у коді.

Було розглянуто можливості використання математичних методів для пошуку оптимальної кількості учасників при заданому обсязі проекту з можливим балансуванням навантаження кожного розробника.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Якубенко, І.М. Agile-менеджмент, як дієве управління проектами для цілеспрямованих команд / Якубенко І.М. – Київ : «Економіка. Менеджмент. Бізнес», 2017. – № 4 (22). – 195 с.
2. Pries, K.H. Scrum Project Management / K.H. Pries, J.M. Quigley. – Boca Raton (Florida): CRC Press, 2010. – 198 p.
3. Кон, М. Scrum: гибкая разработка ПО / Майк Кон – Москва: Вильямс,

2011. – 576 с.

4. Brechner, E. Agile Project Management with Kanban / Eric Brechner. – Microsoft Press, 2015. – 160 p.

УДК 004.032.26 (004.89)

Леощенко С.Д.¹, Олійник А.О.², Субботін С.О.³

¹ асп. НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

³ д-р техн. наук, проф. НУ «Запорізька політехніка»

ВИКОРИСТАННЯ МЕТОДУ КОТЯЧИХ ЗГРАЙ ЯК РОЙОВОЇ ПРОЦЕДУРИ ДЛЯ СИНТЕЗУ НЕЙРОННИХ МЕРЕЖ

Методи штучного інтелекту набули широкого поширення серед завдань як чисто наукових, так і в сфері техніки, бізнесу, фінансів, медичної та технічної діагностики та інших областях, пов'язаних з обробкою інформації. Штучні нейронні мережі (НМ) активно використовуються як для інтелектуального аналізу даних, так і для динамічного аналізу даних, аналізу потоків даних, аналізу великих масивів інформації і т.д.

Для виконання етапу навчання НМ найбільшого поширення набув метод зворотного поширення помилки, який дозволяє підлаштовувати ваги в багатосарових складних НМ за допомогою навчальних наборів. Однак, незважаючи на популярність метод має ряд недоліків:

- досить довгий процес навчання;
- проблема перенавчання мережі;
- суттєва залежність від експерта.

НМ завжди були привабливими з інтуїтивної точки зору. У зв'язку, з чим виникло припущення про те, що для поліпшення роботи може бути доцільно застосувати інше запозичення у природи – еволюційні обчислення.

У наш час досить популярними еволюційними біоінспірованими алгоритмами є, так звані, ройові методи (Particle Swarm Optimization). Алгоритми ройової поведінки запозичені з природи, де групи тварин (зграї, рої) демонструють несподівано гарні результати при тому, що кожна з особин (агентів) має досить примітивний розум. В останні кілька десятиліть в різних областях науки робляться небезуспішні спроби відтворити ройові механізми поведінки в штучних моделях. Багато моделей при цьому прагнуть відтворити властивість, притаманну природним самоорганізаційним системам. В процесі взаємодії своїх елементів система як єдине ціле набуває нових якостей, які не є характерними для окремих її елементів.

Одним з перспективних методів ройового інтелекту є метод оптимізації

на основі котячих зграй (Cat Swarm Optimization – CSO). Це пояснюється достатнім рівнем швидкодії та відносною простотою реалізації. Успішність і перспективність методу також підтверджує ряд модифікацій: процедури, засновані на гармонійному пошуку, дробових похідних, адаптації параметрів пошуку, «crazy cats» та інші.

На рис. 1 наведено загальний алгоритм роботи методу CSO. Слід уточнити, що для застосування методу в якості нейроеволюційного, під кожним агентом (кішкою, особоною) – будемо розуміти окрему нейронну мережу.

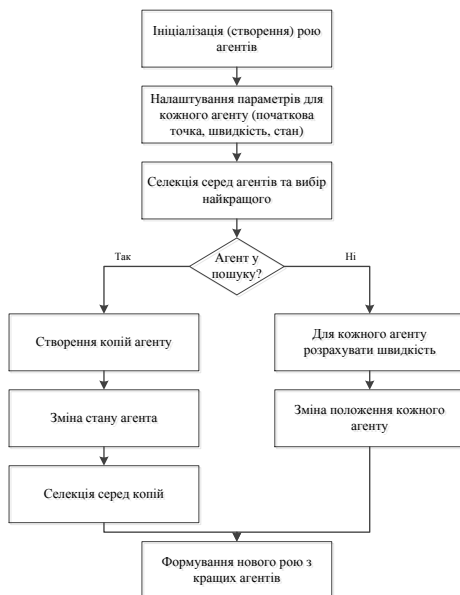


Рисунок 1 – Схематичне представлення роботи методу

Розглянувши метод, можна виділити такі недоліки:

- необхідне багаторазове оцінювання значень цільової функції;
- залежність від пулу (виділеного обсягу пам'яті) під копії особин;
- можлива мала швидкість збіжності.

Це пояснюється тим, що метод пошуку реалізує, по суті, покоординатний спуск (метод Гауса-Зейделя), а в режимі погоні реалізується градієнтний пошук з великим кроком, що в загальному випадку не гарантує відшукування глобального екстремуму. Для усунення зазначених недоліків доцільним є введенням антиградієнтного руху в режимі пошуку, обмеження розміру пулу або використання матриці станів замість зберігання копій.

СТЕНОГРАФІЧНІ МЕТОДИ ЯК ЗАСІБ БОРОТЬБИ З НЕСАНКЦІОНОВАНИМ ДОСТУПОМ

Дуже часто при зберіганні та передачі інформації, а особливо економічної або фінансової інформації, виникає потреба у її приховуванні від сторонніх очей. Тому вусьому світі назріло питання розробки методів (засобів) захисту інформації організаційного, методологічного й технічного характеру, серед них – методи криптографії та стеганографії.

Для гарантованого захисту вмісту повідомлення існує два різних підходи [1-2]: блокування несанкціонованого доступу до інформації шляхом шифрування повідомлення; повідомлення, яке передається, намагаються приховати так, аби його неможливо було знайти. За першим підходом використовують криптографічні методи захисту. Другий підхід застосовує стеганографічні методи захисту. На відміну від криптографічного захисту, стеганографічні методи дозволяють вмонтувати інформацію, що передається, в невинні послання так, щоб не можна було навіть запідозрити існування підтексту. У цьому випадку стеганографія являє собою більш високий рівень захисту інформації в порівнянні з методами криптографії.

Інтерес до стеганографії відродився в останнє десятиліття і був викликаний широким поширенням технологій мультимедіа, що пов'язані із захистом інформації. Ще в 1883 р. фламандський криптограф А. Керхгофс (А. Kerckhoffs) указував на той факт, що система захисту інформації повинна виконувати покладені на неї функції навіть при повній інформованості противника про її структуру та алгоритм функціонування [3]. Вся таємність системи захисту переданих повідомлень повинна втримуватися в ключі – фрагменті інформації, попередньо, як правило, розділеному між адресатами.

Цифрова стеганографія як містить у собі такі напрями: вбудовування інформації з метою її прихованої передачі; вбудовування цифрових водяних знаків (ЦВЗ) (watermarking); вбудовування ідентифікаційних номерів (fingerprinting); вбудовування заголовків (captioning).

ЦВЗ можуть застосовуватися, в основному, для захисту відкопіювання та несанкціонованого використання. Прикладами можуть бути фотографії, аудіо- та відеозаписи та ін. Один з найбільш ефективних технічних засобів захисту мультимедійної інформації полягає у вбудовуванні в захисті об'єкта невидимих міток – ЦВЗ.

Технологія вбудовування ідентифікаційних номерів виробників

має багато загального з технологією ЦВЗ. Відмінність полягає в тому, що впершому випадку кожна захищена копія має свій унікальний вбудований номер (звідси й назва – дослівно "відбитки пальців"). Цей ідентифікаційний номер дозволяє виробникові відслідковувати подальшу долю свого дітища.

Одна із найбільш розвинених на сьогоднішній день група методів вбудовування інформації з метою її прихованої передачі – заміна найменш значущого біта (LSB – LeastSignificantBit) в просторовій множині [3]. Молодший значущий біт зображення несе в собі найменше інформації. Відомо, що людина в більшості випадків не здатна помітити змін у цьому біті. При цьому для зображення в градаціях сірого обсяг убудованих даних може становити 1/8 від загального обсягу контейнера. Популярність цього методу обумовлена його простотою й тим, що він дозволяє приховувати у відносно невеликих файлах досить великі обсяги інформації. Метод найчастіше працює з растровими зображеннями, представленими у форматі без компресії (наприклад: GIF і BMP). Недоліком методу LSB є чутливість до розміру зображення, тобто чим менший розмір зображення, тим більше будуть відрізнятися два сусідні пікселі, тому пропонується використовувати зображення з великою роздільністю. Попри це, метод запису LSB є досить популярним, стійким та простим в реалізації. Існує багато модифікацій цього методу запису, кожна з яких дозволяє корегувати стенографічні процеси в залежності від потреб.

Підводячи підсумок, можна сказати, що використання стенографічного методу LSB для усунення несанкціонованого доступу до даних є ефективним та активно запровадженим рішенням.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Барсуков В.С., Романцов А.П. Комп'ютерна стеганографія вчора, сьогодні, завтра [Електронний ресурс]. – Режим доступу: <http://www.bnti.ru/showart.asp?aid=330&lvl=03.07.06> (дата звернення 03.03.2020). – Назва з екрана.
2. Конахович Г.Ф. Комп'ютерна стеганографія. Теорія і практика [Текст] Г.Ф.Конаховіч, А.Ю. Пузиренко. – К.: МК-Пресс, 2006. – 288 с.
3. Kerckhoffs A. LaCryptographicMilitaire / A. Kerckhoffs/[Journaldessciencesmilitaire]. – 1983. – Jan. – P. 5 – 83; 1983. – Feb. – P. 161–191.

МЕТОДИ АНТИВІДЛАДЖУВАЛЬНОГО ПРОГРАМУВАННЯ

Для створення і розповсюдження конкурентоздатного програмного продукту потрібні чималі фінансові витрати: на розробку, його підтримку і, якщо передбачено, на подальший розвиток. Для окупності витрат зазвичай користувачу надається тимчасово безкоштовна версія продукту, після закінчення якої користувач зобов'язан внести плату за подальше користування. Деякі несвідомі користувачі замість придбання продукту намагаються його несанкціоновано запустити, що ставить під загрозу окупність програмного продукту.

При розробці програмного забезпечення впроваджуються засоби протидії несанкціонованому запуску, такі як: перевірка ключа активації, серійного номеру та інші. Однак ці засоби можуть бути деактивовані через модифікацію програмного коду, який, попередньо, досліджено відладжувачем.

Відповідно, метою впровадження антивідлагоджувальних методів в програмний код є унеможливлення або підвищення складності проведення відлагоджувачем аналізу виконуваного коду і ускладнення відтворення вихідного коду програмного продукту.

Існує дві популярні стратегії протидії відлагоджуванню: впровадження заборони запуску відлагоджувача або виявлення роботи відлагоджувача і подальше переривання нормальної роботи програми.

Заборона запуску базується на блокуванні "відлагоджувальних" переривань процесора (int 1, int 3), що унеможливорює взагалі запуск відлагоджувача. В програму слід впровадити власні обробники відлагоджувальних переривань, адреси яких при запуску повинні бути записані в таблицю векторів переривань процесора. Такі «самописні» обробники переривань можуть складатися з однієї команди – власне ігнорувати команди переривання. Слід врахувати, що можлива ситуація коли зломщик виявить частину коду, що відповідає за запис адреси оброблювачів в таблицю переривань, видалить її, та, відповідно, захист буде деактивовано. Отже, таку частину коду слід уважно шифрувати.

Виявлення роботи відлагоджувача має більш розвинені варіанти розвитку. Одним з таких варіантів є відстежування клавіатури. В процесі трасування програми (при її покроковому виконанні) зломщику необхідно натискати на клавіші для переходу до чергової команди, відповідно, якщо

прибрати реакцію комп'ютера на натискання клавіш - налагодження програми буде зірвано. На початку роботи програми виконується спеціальна команда «CLI», що встановлює прапор ігнорування всіх переривань від зовнішніх пристроїв. У процесі роботи програми відстежується стан цього прапора. Якщо буде використано режим покрокового виконання програми - відладчик дозволить клавіатурні переривання, змінивши при цьому прапор, а, отже, відстежувач зможе зафіксувати відлагоджування. Після виявлення такого режиму можна закрити програму або перейти на хибну гілку виконуваного коду. Слід зауважити, що метод працює тільки в ситуаціях, коли для роботи програми не потрібне введення даних від користувача (оператора).

Іншим методом, схожим до відстежування клавіатури, є слідкування за прапором переривань таймера. Відладчик під час своєї роботи зупиняє програму, а, відповідно, за викликом відповідного переривання, зупиняється таймер виконання програми. Відповідно, досліджуючи стан переривання таймеру, можна виявити роботу роботи відлагоджувача. Ще один варіант з застосуванням таймеру полягає у контролі часу виконання програми. На етапі розробки слід виконати пошук коду програми, який виконується доволі часто з приблизно однаковим часом. Після виявлення таких ділянок коду слід поставити перевірку часу, витраченого на їх виконання. Якщо при виконанні програми на таких ділянках буде виявлено суттєві розбіжності часу – це є свідченням роботи під відлагоджувачем. Методи, які базуються на роботі з таймером, не накладають обмежень на дії користувачів, а, отже, кращі ніж клавіатурний метод.

Проаналізувавши антивідлагоджувальні методи отримано розуміння, що прийоми захисту від відлагоджувачів не є універсальними: їх дієздатність забезпечена лише особливостями архітектури поточних процесорів і наявними підходами відлагоджування програм. Такі особливості обмежені, а їх принципи роботи – загальновідомі.

Однак, незважаючи на це, антивідлагоджувальні методи слід застосовувати при розробці програмного продукту, бо вони здатні забезпечити захист від зломщика-початківця і здатні забрати певний час в професійного зломщика.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Касперські К. Техніка відлагоджування програм без вихідних текстів [Текст] / К. Касперські; – СПб.: БХВ-Петербург, 2005. – 832 с.

ПРИНЦИП БОРОТЬБИ З ПІРАТСТВОМ ВСУЧАСНІЙ ІНДУСТРІЇ ВІДЕОІГОР

У сучасному світі боротьба з піратством є однією з головних проблем при розробці будь-якого програмного забезпечення, у тому числі й відеоігор, від цього залежить дохід з продажу, що на пряму впливає на підтримку поточного проекту та якість розробки наступного.

Боротьба з піратством – це нескінченний і, як правило, безрезультатний для багатьох видавців і розробників процес: якщо люди вирішили грати в гру безкоштовно, то великі шанси, що вони знайдуть спосіб. Але це не означає, що розробники завжди залишатимуться в програші. Багато з них за останні роки використовували досить цікаві і винахідливі методи, щоб спровокувати або познущатися з піратів, які незаконно скачали гру, або навіть змусити їх ненавмисно визнати себе піратам на ігрових форумах на потіху оточуючих [1].

Одним з методів захисту є керування цифровими правами (DRM) яка масово використовується для боротьби з піратами.

DRM (від англ. Digitalrightsmanagement) – програмні або програмно-апаратні засоби, які навмисно обмежують або ускладнюють різні дії з даними в електронній формі (копіювання, модифікацію, перегляд і т.п.), або дозволяють відстежити такі дії. DRM являє собою набір систем контролю і управління доступом. На даний момент DRM використовується безліччю компаній по всьому світу. Сюди входять такі методи захисту як:

1) Активация для обмеження кількості встановлень – у деяких комп'ютерних іграх DRM-захист використовується для обмеження числа систем, на яких можна встановлювати ці програми. Для контролю використовується онлайн-аутентифікація на серверах видавця. Більшість таких DRM-схем дозволяють зробити 3-5 встановлень, проте деякі дозволяють скасувати активацию за допомогою деінсталяції [2].

2) Постійна онлайн-аутентифікація – існує багато відеоігор які використовують подібний метод, зазвичай це онлайн ігри, але зустрічаються й деякі одиночні ігри. Для цього методу зазвичай використовується онлайн лаунчер, через який відбувається придбання ліцензійної копії гри, яка прив'язується до аканту покупця, і яка надалі буде запускатися тільки через цей лаунчер.

3) Втручання в ПО – деякі студії в якості захисту використовують не

зовсім стандартні підходи. Наприклад при запуску нелегальної копії гри використовують DRM-схему яка починає створювати ситуації які просто заважають грати, або роблять подальше проходження неможливим [2].

У DRM є як свої прихильники, так і противники. Перші стверджують, що DRM допомагає боротися з порушеннями авторських прав в мережі і зберігати регулярні надходження доходів. Вони вважають, що подібні цифрові замки необхідні для запобігання крадіжки інтелектуальної власності. Опоненти ж відзначають, що доказів того, що DRM допомагає запобігти порушенню авторських прав, немає. Замість цього DRM тільки приносить незручності законним покупцям і допомагає великим компаніям гальмувати інновації та конкуренцію. Також, багатьом не подобається неавтономні DRM, які вимагають для використання захищеної копії постійне мережеве з'єднання з контролюючою системою [2].

Підводячи підсумок, можна сказати, що на сьогоднішній день є багато різних способів захисту, де DRM лише один з них, деякі розробники ігор створюють свої засоби захисту, які необов'язково будуть заважати грати, але будуть регулярно вносити дискомфорт в ігровий процес. При цьому дана проблема залишається відкритою, тому що яка б складна система захисту не знаходилася в грі, хакери її зламують за дуже короткий період, цей процес можна відстрочити але не вирішити. Єдиний вихід в даній ситуації це перестати користуватися піратськими продуктами.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Кращі ігрові способи покарання піратів [Електронний ресурс]. – Режим доступу: <https://genapilot.ru/best-in-game-piracy-punishments> (дата звернення 04.03.2020). – Назва з екрана.

2. Технічні засоби захисту авторських прав [Електронний ресурс]. – Режим доступу:

https://ru.wikipedia.org/wiki/Технические_средства_защиты_авторских_прав (дата звернення 04.03.2020). – Назва з екрана.

ВИКОРИСТАННЯ ТБВЗ-ТЕХНОЛОГІЙ В РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Теорія вирішення винахідницьких задач (ТБВЗ) – це конструктивна методологія генерації ефективних ідей і вирішення проблем на основі моделей протиріч і методів їх рішення, екстрагованих із відомих прикладів ефективних рішень [1]. Хоча об'єктами класичної ТБВЗ є технічні системи і їх компоненти, проте сьогодні її використовують у різних сферах, у тому числі й у інформаційних технологіях.

Сучасний розвиток сфери інформаційних технологій все більше залежить від якості та вартості програмних продуктів. Однак, розробка якісного програмного забезпечення (ПЗ), в свою чергу, залежить від багатьох факторів. Показники кризи програмування (відставання від графіку, перевищення кошторису, виконання бажань замовника не в повній мірі тощо) добре знайомі всім розробникам великих проєктів. ПЗ володіє такими властивостями як складність, масштабованість, змінність, абстрактність, нематеріальність. Одним з можливих варіантів удосконалення, створення та розвитку програмних продуктів на всіх етапах життєвого циклу можуть бути методи та інструменти ТБВЗ [2].

На сьогоднішній день ТБВЗ активно застосовують в розробці ПЗ: від формування ідеї до експлуатації та супроводження. Серед таких методів та інструментів можна виділити наступні: методи та інструменти розвитку творчої уяви (РТУ), принципи вирішення протиріч, методи пошуку інформації, елєполі, алгоритм вирішення винахідницьких задач (АВВЗ), прогнозування на основі ТБВЗ тощо

На початкових етапах розробки ПЗ доцільним є використання методів та інструментів розвитку творчої уяви для боротьби з інерцією мислення. На початку своєї історії ТБВЗ використовувала метод спроб та помилок, проте цей метод має багато недоліків, що пов'язані з часовими витратами та неефективністю вирішення проблемних задач. Тож було створено інші методи: мозковий штурм, синектика, метод фокальних об'єктів, метод «снігового кому» та «золотої рибки», метод числової осі, метод маленьких чоловічків. Всі ці методи прості для розуміння та використання, не займають багато часу, призводять до появи великої кількості нестандартних ідей та є універсальними. Однак більшість з цих методів не підходять до вирішення задач середньої та високої складності.

Під час розробки програмних продуктів розробники доволі часто стикаються з проблемою виникнення протиріч у завданнях. Для усунення протиріч ТБВЗ пропонує використання таблиці прийомів усунення протиріч, що була розроблена основоположником ТБВЗ Г. С. Альтшуллером. Даний метод є простим у використанні, а підказки, що дає таблиця, допомагають отримати образ рішення задачі. Проте, серед недоліків варто відзначити, що таблиця має обмежений список типових вимог, складність в оновленні та розширенні списку прийомів [2].

Елепольний аналіз – це спосіб дослідження системи, при якому вона поділяється на елементи, що пов'язані полями взаємодії. За допомогою застосування даного аналізу в розробці ПЗ можливо збільшити керованість структури програми, знизити затрати на її написання та використання ресурсів, зробити програму більш надійною. Однак при використанні елепольного аналізу стикаються з тим, що систему складно розділити на елементи та поля – вони взаємопов'язані; не завжди зрозуміло, на якому рівні деталізації потрібно створювати елеполь.

Ключовим інструментом ТБВЗ є АБВЗ. Алгоритм поєднує в собі всі основні інструменти ТБВЗ та організує їх в єдину систему. Метою алгоритму є зниження психологічної інерції, розвиток творчої уяви, виявлення закономірностей розвитку систем. Однак серед основних недоліків можна виділити те, що він є досить важким для оволодіння, потребує постійного тренування.

В якості однієї з особливостей застосування ТБВЗ в розробці ПЗ можна виділити не лише рішення задач, але й виявлення, прогнозування майбутніх можливих задач і протиріч [3]. Для цього існують методи прогнозування засновані на оцінках експертів та екстраполяції. Щодо першого методу, то він надає можливість передбачити важливі повороти в системі, що прогнозується, однак при цьому прогнози будуються на основі суб'єктивного досвіду експертів. Перевагою методу екстраполяції є вивченість моделей, що використовуються, проте прогноз може виявитися помилковим.

Застосування методів та інструментів ТБВЗ в розробці ПЗ стало можливим завдяки адаптації їх для використання в нематеріальних системах. Їх використання в процесі розробки ПЗ дозволяє прийти до кращого рішення проблемних задач, підвищити надійність програми, проаналізувати існуючі рішення, тобто комплексно підійти до рішення поставлених задач.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Орлов М. О. Азбука современной ТРИЗ. Настольная книга для избрательного мышления [Текст]: / М. О. Орлов. – М.: изд. АСТ, 2017. – 495 с.
2. Рубин М. С. Основы ТРИЗ и инновации. Применение ТРИЗ в

программных и информационных системах [Текст]: учебное пособие / М. С. Рубин, В. И. Кияев – М., 2012. – 278 с.

3. ТРИЗ в программировании [Электронный ресурс]. – Режим доступа: <https://scientifically.info/publ/5-1-0-22>.

УДК 303.7

Куковинець О.Є.¹, Зайко Т.А.²

¹ студ. гр. КНТ-127 НУ «Запорізька Політехніка»

² канд. техн. наук, доц. НУ «Запорізька Політехніка»

МЕТОДИ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ АВТОМАТИЗОВАНИХ СИСТЕМ ОБРОБКИ ІНФОРМАЦІЇ В БАНКАХ

Не буде перебільшенням сказати, що загроза умисних порушень функціонування автоматизованих систем обробки інформації в банках (АСОІБ) [1] різного призначення в даний час є однією з найбільших. Найбільш справедливо це твердження для країн, у яких потужна інформаційна інфраструктура, про що переконливо свідчать наведені нижче цифри.

Відомо, що в 1992 році збиток від комп'ютерних злочинів склав \$ 555 млн., 930 років робочого часу і 15.3 року машинного часу. За іншими даними збиток фінансових організацій становить від \$ 173 млн. До \$ 41 млрд. на рік.

З прикладу вище, можна зробити висновок, що системи обробки і захисту інформації відображають традиційний підхід до обчислювальної мережі як до потенційно ненадійному середовищу передачі даних. Існує декілька основних способів забезпечення безпеки програмно-технічного середовища, що реалізуються різними методами.

Перший метод полягає в ідентифікації та авторизації за допомогою паролів. Завдання ідентифікації виконує незалежний сервер, який містить паролі, як для користувачів, так і для кінцевих серверів. Таким чином, використання мережевих послуг вимагає двох паролів, але користувач має знати лише один - другий надається йому сервером «прозорим» чином. Сервер стає вузьким місцем всієї системи, а його злом може негативно вплинути на безпечність всієї обчислювальної мережі. Після вдалої ідентифікації користувач авторизується у системі під своїм обліковим записом.

Другий метод полягає в інкапсуляції переданої інформації в спеціальних протоколах обміну. Використання подібних методів в комунікаціях засноване на алгоритмах шифрування з відкритим ключем. На етапі ініціалізації відбувається створення пари ключів - відкритого і

закритого, наявного тільки у того, хто публікує відкритий ключ. Ідея алгоритмів шифрування з відкритим ключем полягає в тому, що операції шифрування і дешифрування виконуються різними ключами (відкритим і закритим відповідно).

Третій метод полягає в обмеженні інформаційних потоків. Це відомі технічні прийоми, що дозволяють розділити локальну мережу на пов'язані підмережі і здійснювати контроль і обмеження передачі інформації між цими підмережами. До таких прийомів відносять Firewalls (брандмауери) та Proxyservers.

Четвертий метод полягає у створенні віртуальних приватних мереж (VPN) [2], що дозволяють ефективно забезпечувати конфіденційність інформації, її захист від прослуховування або перешкод при передачі даних. Вони дозволяють встановити конфіденційний захищений зв'язок у відкритій мережі, якою зазвичай є інтернет і розширювати межі корпоративних мереж до віддалених офісів, мобільних користувачів, домашніх користувачів та партнерів по бізнесу.

П'ятий метод полягає у впровадженні систем виявлення вторгнень і сканерів уразливості, які створюють додатковий рівень мережевої безпеки.

Система виявлення вторгнень Cisco Intrusion Detection System (IDS) [3] може захистити мережу по периметру, мережі взаємодії з бізнес-партнерами і все більш вразливі внутрішні мережі в режимі реального часу. Система використовує агенти, що представляють собою високопродуктивні мережеві пристрої, для аналізу окремих пакетів з метою виявлення підозрілої активності.

Cisco Secure Scanner [3] являє собою програмний сканер промислового рівня, що дозволяє адміністратору виявляти і усувати уразливості в мережевій безпеці перш, ніж їх знайдуть хакери.

Підсумовуючи наведені методи, можна сказати, що розробка інформаційних систем вимагає паралельної розробки технологій передачі та захисту інформації. Ці технології повинні забезпечувати захист інформації, що передається, роблячи мережу «надійною», хоча надійність на сучасному етапі розуміється як надійність не на фізичному рівні, а скоріше на логічному (інформаційному рівні).

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Демин В.С. Автоматизированные банковские системы [Текст] / В.С. Демин. – М: Менатеп-Информ, 1997. – 302 с.
2. Гайкович Ю.В. Безопасность электронных банковских систем [Текст] / Ю.В. Гайкович, А.С. Першин. – М: Единая Европа, 1994. – 605 с.
3. Загрози безпеці автоматизованих систем: [Електрон. ресурс]. – Режим доступу: <http://www.modestbank.ru/bksys-752-1.html>

ІНФОРМАЦІЙНИЙ ЗАХИСТ В ІНДУСТРІЇ ВІДЕОІГОР

Ігрові сервіси – це основний спосіб отримання ігор та контенту. У домашніх консолях це сервіси від виробників пристрою, на зразок Xbox Store і PS Store, на персональних комп'ютерах є Steam, Epic Games Store, Origin, Uplay, GOG, Battle.net, а у випадку з мобільними пристроями - Google Play і App Store [1].

Найпопулярнішим на персональних комп'ютерах є сервіс компанії Valve Steam. Корпорація Valve є відомим у світі розробником відеоігор, а також спеціалізується на цифровій дистрибуції. Сервіс Steam надає послуги цифрової дистрибуції, багатокористувацьких ігор і спілкування гравців. Через Steam поширюється близько 23000 продуктів, кількість активних користувачів перевищує 125 мільйонів, щодня сервісом у середньому користуються 14 мільйонів осіб. По оцінці експертів сервіс Steam охоплює 70% ринку цифрової дистрибуції відеоігор у світі.

9 грудня 2018 р. компанія Valve обмежила обмін предметами в сервісі Steam для користувачів, які не використовують двоетапну перевірку через мобільний додаток. Щоб донести до геймерів важливість застосування цього методу, компанія привела тривожну статистику злому облікових записів і докладніше розповіла про причини свого рішення. З'ясувалося, що щомісяця в руки зломисників потрапляють близько 77 тис. аккаунтів, і кількість зломів продовжує зростати.

Існує два варіанти викрадення інформації:

1) За допомогою програми, яка відсилає з Вашого комп'ютера інформацію зломисникові. Говорячи простіше це вірус (шпигун, кейлогер, троян та інші).

2) Користувач добровільно передає інформацію.

Зазвичай таким способом є фішингові сайти. Це саме ті сайти, які мають ідентичний вигляд оригінального сайту, але з іншою адресою (доменом). Адреса буває дуже схожою на адресу необхідного вам сайту, але візуально його можна прийняти за справжній. Відмінність може бути в одній букві по типу: mn, nn, mm або nm від повної адреси [2].

Такі фішингові сайти могли прислати ваші недавно додані друзі або зломисник який вкрав аккаунт вашого друга і пише під його виглядом. Частіше всього такі сайти присилають з проханням проголосувати або з наданням можливості отримати гру або внутрішньоігрові речі безкоштовно

або недорого.

Бувають більш продумані ходи з боку викрадача інформації.

Він може спочатку вистежувати Вашу мережеву активність завдяки можливостям Steam і подружитися під час гри, наприклад.

Далі почне вивчати Ваш список друзів і аналізувати активність спілкування.

Після цього, в один прекрасний момент, він повністю переробить свій профіль під профіль Вашого друга (будуть максимально схоже збігатися імена, аватарки і інша інформація про аккаунт) та піде з Вами на контакт [3].

Щоб захистити свій обліковий запис Steam доцільно використовувати Steam Guard. Steam Guard - це додатковий рівень безпеки, який може бути використаний на ваш обліковий запис Steam. Перший рівень безпеки - ваші облікові дані: логін аккаунта і пароль. Активована функція Steam Guard ускладнить доступ до аккаунту для сторонніх осіб. Якщо на аккаунті активована функція Steam Guard, для входу в нього з неавторизованого пристрою буде потрібно спеціальний код доступу. Залежно від ваших налаштувань Steam Guard, код доступу ви отримаєте або в повідомленні, надісланому на контактну адресу електронної пошти, або через мобільний додаток Steam [2].

Також є додатковий спосіб захисту Сімейний перегляд, який більше підходить для контролю дітей, але підійде також і для захисту. Для користування ним треба створити додаткову пошту та придумати 4-значний пароль. При вході в Steam буде доступна тільки бібліотека користувача з такою кількістю ігор яку він вказав при налаштуванні. Можна вказати як всі ігри так і ні одну, що буде краще для захисту. Щоб мати можливість користуватися треба вийти з Сімейного перегляду увівши придуманий пароль.

Популярність Steam стрімко зростає, а це значить, що будуть з'являтися нові способи обману користувачів. Не виключено, що це буде шкідливий код, який Ви по необережності знайдете на просторах Інтернету. Він же в свою чергу вчепитися в браузер і надалі буде переадресовувати на фішингові сторінки [3].

В більшості інших сервісів теж є додаткові захисні функції. Наприклад двоетапна аутентифікація через пошту або через Google Authenticator як в Uplay.

Отже сьогодні інформаційний захист персональних даних в ігровій індустрії залишається відкритою. Навіть якщо користуватися всіма перерахованими вище методами захисту це на 100% не захистить аккаунт від злому та шахрайства. Тому потрібно бути дуже обережним реєструючись на різних сайтах через аккаунт Steam або аккаунти інших сервісів та ретельно перевіряти такі сайти на дійсність та чесність.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Відеоігри та інформаційна безпека: як програти [Електронний ресурс]. – Режим доступу:

<https://www.securitylab.ru/blog/company/falcongaze/338191.php> (дата звернення – 05.03.2020). – Назва з екрана.

2. Безпека соціально-економічних процесів в кіберпросторі [Електронний ресурс]. – Режим доступу:

<https://knute.edu.ua/file/NjY4NQ==/250dafc576ffd3c6a92546eebacc834d.pdf> (дата звернення - 05.03.2020). – Назва з екрана.

3. Як захистити свій Steam аккаунт [Електронний ресурс]. – Режим доступу: <https://steamcommunity.com/sharedfiles/filedetails/?id=252672665> (дата звернення – 05.03.2020). – Назва з екрана.

УДК 004.056.5

Зайко Т.А.¹, Козлов В.В.²

¹ канд. техн. наук, доц. НУ «Запорізька Політехніка»

² студ. гр. КНТ-137 НУ «Запорізька Політехніка»

ОБФУСКАЦІЯ ЯК ЗАСІБ ЗАХИСТУ ВИХІДНОГО КОДУ ПРОГРАМИ

Сьогодні інформацію розглядають як один з основних ресурсів розвитку суспільства, а інформаційні системи і технології як засіб підвищення продуктивності та ефективності роботи людини. Тому інформація та інформаційні технології є найціннішим і найдорожчим ресурсом, який потребує захисту.

Найбільш серйозна проблема при захисті програмного забезпечення – це протидія різним засобам статичного і динамічного аналізу коду, які використовуються для реверсінга.

Реверсінг, реверс-інжинірінг або зворотна розробка – це дослідження деякого пристрою або програми, а також документації на них з метою зрозуміти принцип і алгоритм його роботи. Під час зворотної розробки дослідник вивчає програму з закритим вихідним кодом, аналізує її структуру та принцип роботи, можливо, вносить якісь зміни [1].

Мета розробника захисту полягає в тому, щоб якомога сильніше ускладнити і затягнути роботу по реверсінгу.

Для цього використовують процес засмічення коду або обфускації. Обфускація – це приведення виконуваного коду або вихідного тексту програми до виду, який зберігає її функціональність, але ускладнює розуміння і аналіз алгоритмів його роботи.

Обфускацію використовують для того, щоб ускладнити

процесдекомпіляції і вивчення програми [2], запобігти обхід систем перевірки ліцензій та зменшити розмір працюючого коду, а отже і прискорити його роботу.

Існує три основних технології, які використовуються в залежності від мови програмування і способу розповсюдження програми.

Перша технологія – це технологія, що використовується на рівні вихідних текстів програми. На JavaScript, VBScript та подібних скрипт-мовах вихідний текст завжди доступний користувачу. В цьому випадку форматуванням тексту і заміною імен змінних можна зробити вихідний текст програми менш читабельним. Найпопулярніший варіант на даний момент – це перейменування в недруковані символи або використання коротких незрозумілих ідентифікаторів. Також одним з підходів є створення великої кількості *overload* методів з одним ім'ям, які мали до обфускації різні імена, і ніяк не були пов'язані.

Друга технологія – це технологія, яка використовується на рівні проміжного або байт-коду. Java і мови платформи .NET компілюють вихідний код в байт-код, що містить достатньо інформації для того, щоб відновити вихідний код. Для таких мов використовується принцип зміни потоків управління. На цьому етапі змінюються інструкції та їх порядок в коді, вводяться умовні оператори, а методи розбивають на блоки.

Третя технологія – це технологія, яка використовується на рівні машинного коду. Мови програмування, такі як C++ та Pascal, компілюють вихідний код в машинний. В цьому випадку обфускація застосовується в критичних до безпеки, але не критичних до швидкості місцях, шляхом вставки недіючих конструкцій.

Представлені технології прийнято використовувати як в сукупності, так і окремо [3]. Все залежить від бажаного ступеня захисту програми і мети його розробки. Найчастіше всі дії зводяться до того, що в програму вносяться спеціальні функції, які виконують складні дії, звертаються до накопичувачів даних, але нічого не змінюють.

Обфускація докучає хакерам, перешкоджаючи реконструкції алгоритмів і швидкому злому захистів, проте і створює проблеми в антивірусній індустрії. Щоб зламати програму, аналізувати її алгоритм не обов'язково. А ось виявити шкідливий код (він же *malware*) без цього вже не вдасться. Також до основних недоліків використання обфускації можна віднести ще втрату гнучкості коду і труднощі його налагодження.

Підводячи підсумки можна сказати, що обфускація допомагає зробити розподілену систему безпечнішою, але не варто обмежуватися тільки нею. Обфускація – це безпека через неясність. Жоден з існуючих алгоритмів обфускації не гарантує складності декомпіляції і не забезпечує безпеки на рівні сучасних криптографічних схем. Тому її слід проводити в комплексі з

іншими методами шифрування і захисту даних.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Панов А.С. Реверсинг и защита программ от взлома / А.С. Панов. – Спб.: БХВ-Петербург, 2006. – 256 с.
2. Brunton F. Obfuscation: A User's Guide for Privacy and Protest / F. Brunton, H. Nissenbaum. – Massachusetts.: Press Cambridge, 2015. – 107 p.
3. Оголюк А.А. Защита приложений от модификации. Дополнительные материалы / А.А. Оголюк. – Спб.: СПбГУ ИТМО, 2014. – 122 с.

УДК 303.7

Марков О. С.¹, Зайко Т. А.²

¹студ. гр. КНТ-117 НУ «Запорізька Політехніка»

²канд. техн. наук, доц. НУ «Запорізька Політехніка»

РЕАЛІЗАЦІЯ УСКЛАДНЕННЯ ЛОГІКИ ЯК МЕТОД ЗАХИСТУ ВІД ДИЗАСЕМБЛЮВАННЯ

Не буде перебільшенням констатувати, що механізм дизасемблювання (статичного аналізу коду) таїть в собі велику небезпеку для переважної більшості продуктів програмного забезпечення (ПЗ). На думку Є. Касперського, фахівця кібербезпеки світового рівня, сучасні версії дизасемблерів можуть виявляти практично будь-який захист [1]. Однак це в жодному разі не є спростуванням методів та прийомів захисту від дизасемблювання кодів.

Важливим для розуміння процесу дизасемблювання, є той факт, що його виконання – процес надзвичайно кропіткий, перш за все, через складність розпізнавання коду. Так, дизасемблювання програми-хробака Морріса, що інфікував 6 200 комп'ютерів [2], спричинило застій в роботі мережі, збитки у часі від якого вимірювались мільйонами доларів США. Таким чином, умисне ускладнення логіки на декілька порядків програмістами робить зворотну інженерію і модифікацію тексту програми завданням неймовірно завищеної вартості.

Одним із найбільш ефективних засобів захисту кодів від дизасемблювання є їхнє ускладнення, або, висловлюючись неформально, маскуванню програми. Воно полягає в багатокроковій обфускації графа потоку керування програмою (ГПКП). Проте слід враховувати, що при застосовуванні комбінованих чи нестандартних методів маскуванню, об'єми програми мають вкладатися в межі обчислювальних систем [3].

Ідею ускладнення логіки можна виразити за допомогою таких

положень. По-перше, усі дуги графа потоку керування, які генеруються при маскуванні ГПКП, спричиняють стрімкий ріст часу роботи програми зломисника, яка передусім спрямована на зняття захисту. По-друге, складність мовних конструкцій (як-от, безлічі непрозорих предикатів), що наявні у замаскованих функціях програми, нерідко зводять нанівець спроби проаналізувати та виділити «холостий» код програми.

Таким чином, реалізація вищенаведених положень при маскуванні у продуктах ПЗ суттєво підвищує рівень захищеності кодів. В процесі маскування програм виділяються декілька ідейних етапів, що можуть реалізовуватися різними програмними методами.

На першому етапі виконується збільшення розміру графа потоку управління функції. Даний крок маскування полягає у перетвореннях структури циклів функцій програми, що в найбільш примітивній реалізації виражається у клонуванні базових блоків. Більш складними засобами реалізації є переплетення окремих функцій, винесення груп операторів, вставлення функцій тощо.

На другому етапі відбувається руйнування структури ГПКП шляхом внесення в нього нових дуг та подрібнення базових блоків з додаванням нових, тимчасово не заповнених. Ціллю цього етапу є підготовка точок програми, куди в майбутньому буде внесено недосяжний, мертвий або надлишковий код.

На наступному етапі здійснюється заповнення заздалегідь підготовлених порожніх блоків інструкціями, які не впливають на результат роботи програми.

Останній етап маскування полягає у поєднанні «холостого» коду з основним. Інструментами реалізації цього етапу є, перш за все, засоби мов програмування, що використовуються, наприклад, покажчики на блоки пам'яті, логічні, відносні та умовні оператори.

Варто зазначити, що описані етапи маскування не є неухильним алгоритмом ускладнення логіки, але їх покрокова реалізація збільшить ймовірність збереження даних щонайменш від програм зломисників-любителів.

На сьогодні, існує велика кількість ефективних (здебільшого, комерційних) дизасемблерів. Теперішніми фаворитами даної галузі можна вважати IDA Assembler та фреймворк Radare2 [4]. IDA містить вбудований відлагоджувач, декомпілятор коду застосунків на С, що дозволяє швидко аналізувати програми (на щастя, професійна версія ліцензії IDA доступна вузькому колу осіб). Radare2 є безкоштовним продуктом, що підтримується спільною розробників; крім відлагоджувача, він також містить декомпілятор, hex-редактор, утиліти порівняння бінарних файлів тощо.

Підсумовуючи наведене, можна сказати, що боротьба проти

зловмисного дизасемблювання відбувається передусім на логічному (інформаційному), а не на фізичному рівні, що передбачає неосязний простір для фантазії програмістів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Касперски К. Техника и философия хакерских атак / К. Касперски. – М.: Солон, 1999. – 272 с.
2. Хробак Морріса: [Електрон. ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Хробак_Моріса
3. Захист програмного забезпечення. Частина 2 : навчальний посібник / В. А. Каплун, О. В. Дмитришин, Ю. В. Барішев. – Вінниця : ВНТУ, 2014. – 105 с.
4. Выбираем инструменты для реверса: [Електрон. ресурс]. – Режим доступу: <https://hacker.ru/2018/12/04/apps4hack/>

УДК 303.7

Марічев Д.О.¹, Зайко Т.А.²

¹ студ. гр. КНТ-117 НУ «Запорізька Політехніка»

² канд. техн. наук, доц. НУ «Запорізька Політехніка»

ПРИНЦИП РОБОТИ SSL СЕРТИФІКАТУ ШИФРУВАННЯ В HTTPS ПРОТОКОЛІ

При створенні програмного забезпечення (ПЗ), а саме веб-сайтів, розробники стикаються з проблемою незахищеності даних при їх передачі від клієнту до серверу та навпаки. Через це під час передачі є ймовірність перехоплення та змінення цінної інформації шахраями. Обов'язкове шифрування вимагає вся інформація, що стосується проведення платежів в інтернеті: оплата товарів в інтернет-магазинах будь-яким способом, оплата послуг через інтернет-банкінг, виконання платежів в онлайн сервісах і багато іншого [1].

Для вирішення проблеми перехоплення цінної інформації все більше сайтів переходять на захищений протокол HTTPS. У протоколі безпеки HTTPS використовується асиметрична схема шифрування за рахунок використання SSL сертифікату шифрування – завдяки цьому всі дані надійно захищені від перехоплення.

В основі будь-якого методу шифрування лежить ключ. Ключ – це спосіб зашифрувати або розшифрувати повідомлення. В роботі SSL сертифіката беруть участь три ключі: публічний, приватний і сеансовий.

Публічний ключ зашифровує повідомлення. Браузер використовує його,

коли потрібно відправити призначені для користувача дані серверу. Наприклад, після того як було введено дані банківської картки і натиснуто «Оплатити». Цей ключ видно всім, браузер прикріплює його до повідомлення.

Приватний ключ розшифровує повідомлення. Його використовує сервер, коли отримує повідомлення від браузера. Цей ключ зберігається на сервері і ніколи не передається разом з повідомленням.

Сеансовий ключ одночасно зашифровує і розшифровує повідомлення. Браузер генерує його на час, який користувач проводить на сайті. Варто користувачеві закрити вкладку, сеанс закінчиться і ключ перестане працювати.

Шифрування з двома різними ключами називають асиметричним. Використовувати такий метод більш безпечно, але повільно. Тому браузер і сервер використовують його один раз: щоб створити сеансовий ключ.

Шифрування з одним ключем називають симетричним. Цей метод зручний, але не такий безпечний. Тому браузер і робить унікальний ключ для кожного сеансу замість того, щоб зберігати його на сервері.

Браузер і сервер встановлюють SSL з'єднання кожного разу, коли користувач заходить на сайт. Це займає кілька секунд під час завантаження сайту.

Коли вводять адресу сайту в браузері – він запитує у сервера, встановлений для сайту сертифікат. У відповідь сервер відправляє загальну інформацію про SSL сертифікат і публічний ключ. Браузер звіряє інформацію зі списком авторизованих центрів сертифікації. Такий список є у всіх популярних браузерах. Якщо все в порядку, браузер генерує сеансовий ключ, зашифровує його публічним ключем і відправляє на сервер. Сервер розшифровує повідомлення і зберігає сеансовий ключ. Після цього між браузером і сайтом встановлюється безпечне з'єднання через протокол HTTPS [2].

Всі сучасні браузери підтримують захищений протокол HTTPS. І все більше сайтів переходять на його використання – особливо після того, як компанія Google оголосила використання зашифрованого протоколу фактором ранжирування при видачі результатів пошуку.

3 січня 2017 року компанія Google почала позначати сайти, які не працюють за HTTPS протоколі як ненадійні. Це означає, що в браузері Google Chrome з'явиться відмітка для сайтів без SSL сертифіката. Якщо на сайті є контактні форми, але немає сертифіката або він встановлений з помилкою – поруч з адресою сторінки з'явиться слово «ненадійних». На усіх сайтах, де встановлений SSL сертифікат, відображається слово «Надійний» [3].

У квітні 2019 року стало відомо про те, що 20% найбільших в світі

сайтів не використовують протокол HTTPS з підтримкою шифрування, незважаючи на обмеження, з якими вони стикаються у зв'язку з цим.

За даними компанії Google, 79 зі 100 найбільш популярних веб-ресурсів, які не пов'язані з компанією, не використовують сертифікат для захищених HTTPS з'єднань. Таку безпеку ігнорують, зокрема, найбільша в світі база даних і інтернет-портал про кінематограф IMDb і газета The New York Times [4].

Таким чином, можна зробити висновок, що використання HTTPS протоколу є майже необхідним у сучасних реаліях, так як він дозволяє захищати персональні дані користувачів від перехоплення та змінення шахраями; допомагає просуванню сайту у ранжуванні в пошуковій видачі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Що таке HTTPS та для чого його використовувати: [Електрон. ресурс]. – Режим доступу: <https://www.kasper.by/blog/chto-takoe-https-i-dlya-chego-ego-ispolzovat/>

2. Як працює SSL сертифікат: [Електрон. ресурс]. – Режим доступу: <https://ssl.com.ua/info/how-ssl-works/>

3. Google: як SSL сертифікат впливає на SEO: [Електрон. ресурс]. – Режим доступу: <https://ssl.com.ua/info/ssl-and-google-seo/>

4. Переваги HTTPS над HTTP: [Електрон. ресурс]. – Режим доступу: http://www.tadviser.ru/index.php/Статья:HTTP_-_HTTPS

УДК 303.7

Сушевських О. В.¹, Зайко Т. А.²

¹ студ. гр. КНТ-117 НУ «Запорізька Політехніка»

² канд. техн. наук, доц. НУ «Запорізька Політехніка»

СУЧАСНІ ТЕХНОЛОГІЇ ДАМПІНГА І ЗАХИСТУ ВІД НЬОГО

Як відомо, одним із найпоширеніших механізмів захисту ПЗ є використання навісного захисту. Основна ідея такого захисту – ускладнити аналіз роботи програми за допомогою шифрування коду програми і розшифровування його безпосередньо перед виконанням. Такий захист забезпечує відносно стійкий та дешевий захист. При його використанні оригінальний код програми шифрується, модифікується РЕ-заголовок, до програми додається розшифровувач. Перед виконанням програма розшифровується, частково відновлюється оригінальний РЕ-заголовок, і керування передається програмі. Але такий захист, як і будь-який інший, не є гарантією того, що програма буде повністю і назавжди захищеною. І одним з

методів зняття захистів такого виду є використання дамперів, тобто програм, які можуть зберегти дамп пам'яті під час роботи програми в момент закінчення спрацьовування захисту.

Зняття такого захисту проходить в три етапи:

- знаходження моменту передачі керування оригінальній програмі;
- знімання дампу пам'яті програми;
- відновлення PE-заголовку.

З цього можна зробити такий висновок: посилити захист можна, максимально ускладнивши для хакера виконання цих операцій. Оскільки предметом даного розділу є дослідження можливості зняття образу програми з пам'яті.

Дамп пам'яті – це копія вмісту оперативної пам'яті, що знаходиться на жорсткому диску або іншому енергонезалежному пристрої пам'яті [1].

Всі дамperi процесів побудовані на функціях `OpenProcess / ReadProcessMemory / VirtualQueryEx` e.t.c. Для отримання списку модулів завантажених в процес зазвичай використовуються функції `ToolHelp API`, які в свою чергу читають пам'ять процесу через `ReadProcessMemory`. На рівні `NativeAPI` при цьому відбувається виклик функцій `ZwOpenProcess` і `ZwReadVirtualMemory`. Очевидний спосіб протидії дампу - це встановити драйвер, який перехопить в ядрі ці функції і заборонить доступ до захищеного процесу.

Все добре в теорії, але при практичній реалізації цього методу зустрічаються підводні камені. Наприклад, в `Windows XP` існує служба стилів. Для правильної її роботи потрібно дозволяти доступ процесу сервера підсистеми (`csrss.exe`) до пам'яті захищеного процесу.

Захист в нульовому кільці відкриває дійсно багаті можливості (обмежені тільки уявою). Наприклад вельми непоганий спосіб антидампа - руйнування таблиці сторінок захищеного процесу. Для цього треба втрутитися в роботу планувальника і перехопити функцію яку не можна експортувати (`SwapContext`), яка викликається при зміні робочого потоку, в обробнику перехоплення, при перемиканні на захищений процес потрібно відновлювати таблицю сторінок, а при відключенні від нього - руйнувати. Це найпростіше, що можна протиставити драйверному дампу. [2]

Інший розповсюджений метод захисту від зняття дампу – динамічне розпаковування. Суть його в тому, що протектор розпаковує захищену програму не повністю, а частинами. Спочатку розпаковується перша сторінка, а коли відбувається звернення за її межі, протектор перехоплює виключення і розпаковує запитану сторінку, при цьому він може прибрати з пам'яті попередню сторінку. Таким чином, образ захищеного процесу в пам'яті ніколи не існує повністю, отже звичайним дампером його не зняти не можна. Для зняття цього захисту більшість хакерів обирають не зовсім

оптимальний шлях – реверсинг і зміну коду протектора з метою змусити його розшифрувати код повністю. Це дуже трудомістка операція, тим більш, що в кожній новій версії автори міняють реалізацію цього захисного механізму, і тоді старі програми для зламу стають неактуальними.

Підводячи підсумок, можна сказати, що метод динамічного розпаковування, є кращим захистом з перерахованих методів та активно запровадженням рішенням.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1 Каплун, В. А. Захист програмного забезпечення. Частина 2 : навчальний посібник / В. А. Каплун, О. В. Дмитришин, Ю. В. Барішев. – Вінниця : ВНТУ, 2014. – 105 с.

2. Современные технологии дампинга и защиты от него. <https://wasm.in/blogs/sovremennye-technologii-dampinga-i-zaschity-ot-nego.396/>(Електронний ресурс)

УДК 004.056.5

Зосенко К. В.¹, Зайко Т. А.²

¹студ. гр. КНТ-127 НУ «Запорізька Політехніка»

² канд. техн. наук, доц. НУ «Запорізька Політехніка»

ІНСТРУМЕНТИ ЗЛОМУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У процесі свого розвитку програмне забезпечення стало виходити за рамки технічних пристроїв і стало проникати в різні сфери людської діяльності. Програмне забезпечення управляє каналами постачання, надає доступ до глобальної інформації, дозволяє управляти заводами і фабриками і використовується для взаємодії з замовниками. Будь-яка помилка в такому програмному забезпеченні може призвести до тяжких наслідків. І саме такими помилками ПЗ користуються сучасні злочинці.

Злом програмного забезпечення був зведений в ранг мистецтва, це дійсно непросте завдання. Спочатку потрібно зрозуміти, яке завдання вирішує фрагмент коду. Часто це можна зробити тільки за результатами роботи. Іноді програмний код можна розділити на кілька фрагментів і вивчити їх окремо. Іноді призначення програмного коду визначається за допомогою некоректних вхідних даних. Цей код можна дизасемблювати і декомпілювати. Іноді з його допомогою код можна аналізувати або вивчити проект програми і архітектурні проблеми. Існує кілька типів програм, активно застосовуються для зламу захисту програми.

Відладчики і дизасемблери традиційно використовуються в парі,

оскільки дизасемблер видає лише «чистий код», хоча сучасні дизасемблери здатні також розпізнавати виклики стандартних функцій, виділяти локальні змінні в процедурах і надавати інші подібні послуги. Користуючись дизасемблером, можна лише здогадуватися, які дані отримує та чи інша функція в якості параметрів і що вони означають. Щоб з'ясувати це, частіше за все потрібно вивчення якщо не всієї програми, то досить значної її частини.

Відладчики виконують інші функції: вони дозволяють аналізувати код в процесі його роботи, відстежувати і змінювати стан регістрів і стека, правити код на льоту - в загальному, спостерігати за діями програми і навіть активно в них втручатися.

Декомпілятори і вузькоспеціалізовані отладчики. З ростом потужності ПК широкого поширення набули компілятори, що створюють не "чистий" машинний код, а якийсь набір умовних інструкцій, який виконується за допомогою інтерпретатора.

Найчастіше буває потрібно дізнатися, які саме дії виконує програма, звідки вона читає і записує дані, які стандартні функції і з якими параметрами викликає. Отримати ці відомості допомагають утиліти моніторингу та API-шпигуни. Вони діляться на дві великі групи: відстежують сам факт виникнення будь-яких подій і дозволяють виявити один або кілька специфічних типів змін, що відбулися в системі за певний проміжок часу. Основна проблема при роботі з такими утилітами добре описується афоризмом: «Після - не означає внаслідок».

Існує також величезна кількість утиліт, які не вписуються в розглянуті вище категорії або потрапляють в кілька категорій відразу. Більш того, деякі утиліти, які можуть бути корисні для злому, створювалися для зовсім інших цілей. Як неможливо досягнути неосяжного, так і не можна описати всі програми, які застосовуються для злому.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Росс Джей Андерсон Інженерія безпеки. Керівництво по створенню надійних розподілених систем / Ross J. Anderson, Cambridge, John Wiley & Sons, 2001. – 640 р.

РОЗРОБКА І РЕАЛІЗАЦІЯ ВБУДОВАНОГО ЗАХИСТУ ПРОГРАМ

Комерційні програми зазвичай захищають від несанкціонованого тиражування. Наявність доступу тільки до носія інформації з дистрибутивом (набором інсталяційних файлів) програмного продукту не повинна давати можливості встановити працездатну копію програми. Тобто даних дистрибутива, який можна скопіювати або непомітно взяти на декілька днів, не повинно вистачати для створення працездатної копії програми. Подібні обмеження можуть бути реалізовані різними способами. Наприклад, дуже багато комерційних програм при інсталяції вимагають ввести серійний номер, надрукований на коробці або вказаний в одному з документів, що додаються до програмного продукту (у Microsoft – в сертифікаті автентичності) [1].

Залежно від можливих порушень у роботі системи та загроз несанкціонованого доступу до інформації численні види захисту можна об'єднати у такі групи: моральноетичні, правові, адміністративні (організаційні), технічні (фізичні), програмні. Зазначимо, що такий поділ є досить умовним. Зокрема, сучасні технології розвиваються в напрямку сполучення програмних та апаратних засобів захисту [2].

Криптографічний захист (шифрування) інформації – це вид захисту, який реалізується за допомогою перетворень інформації з використанням спеціальних (ключових) даних з метою приховування змісту інформації, підтвердження її справжності, цілісності, авторства тощо. На відміну від тайнопису, який приховує сам факт передавання повідомлення, зашифровані повідомлення передаються відкрито, приховується їхній зміст.

Методи криптографії поділяють на дві групи – підставлення (заміни) і переставлення. Підстановчий метод передбачає, що кожна літера та цифра повідомлення замінюється за певним правилом на інший символ. Зокрема, для визначення порядку підставлення може використовуватись певне слово або фраза – ключ. У загальному випадку, у криптографії ключ – це послідовність бітів, що використовуються для шифрування та розшифрування даних.

Розглядаючи програмні засоби захисту, доцільно спинитись на стеганографічних методах. Слово «стеганографія» означає приховане письмо, яке не дає можливості сторонній особі взнати про його існування. Одна з перших згадок про застосування тайнопису датується V століттям до н. е.

Сучасним прикладом є випадок роздрукування на ЕОМ контрактів з малопомітними викривленнями обрисів окремих символів тексту – так вносилась шифрована інформація про умови складання контракту.

Комп'ютерна стеганографія базується на двох принципах. По-перше, аудіо- і відеофайли, а також файли з оцифрованими зображеннями можна деякою мірою змінити без втрати функціональності. По-друге, можливості людини розрізняти дрібні зміни кольору або звуку обмежені. Найчастіше стеганографія використовується для створення цифрових водяних знаків. На відміну від звичайних їх можна нанести і відшукати тільки за допомогою спеціального програмного забезпечення – цифрові водяні знаки записуються як псевдовипадкові послідовності шумових сигналів, згенерованих на основі секретних ключів. Такі знаки можуть забезпечити автентичність або недоторканість документа, ідентифікувати автора або власника, перевірити права дистриб'ютора або користувача, навіть якщо файл був оброблений або спотворений.

Криптографічні алгоритми використовуються як для шифрування повідомлень, так і для створення електронних (цифрових) підписів (ЦП) – сукупностей даних, які дають змогу підтвердити цілісність електронного документа та ідентифікувати особу, що його підписала. Звичайно терміни «електронний підпис» і «цифровий підпис» застосовуються як синоніми, але перший з них має ширше значення, оскільки позначає будь-який підпис в електронній формі («оцифрований» не означає «цифровий»). Отже, електронні підписи не обов'язково базуються на криптографічних методах і можуть бути створені, наприклад, за допомогою засобів біометрії[3].

Таким чином, можна зробити висновок, що використання принципів вбудованого захисту інформації для захисту програм від хакерів у сфері розробки ПЗ є одним з найефективнішим та активно запровадженим рішенням.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Шевчук Р.П. Опорний конспект лекцій з дисципліни „Методи та засоби захисту програмного забезпечення”, для студентів напрямку „Комп'ютерні науки” / Р.П. Шевчук. – Тернопіль, 2007. – 50 с.
2. Мандиа К. Защита от вторжений. Расследование компьютерных преступлений / К. Мандиа, К. Просис.– М., 2005. – 496 с.
3. Луцкер А. Авторское право в цифровых технологиях и СМИ / А. Луцкер. – М., 2005. – 416 с.

БЕЗПЕКА ВЕБ-ДОДАТКІВ. МІЖСАЙТОВИЙ СКРИПТИНГ (XSS) ТА МЕТОДИ БОРОТЬБИ З НИМ

Сьогодні є нормальним, коли через Інтернет люди ведуть приватне листування, оформляють банківські перекази чи зберігають у “хмарі” персональні дані. Більшість користувачів вже звикли довіряти веб-сервісам. Тим не менш, в мережі Інтернет ще є багато сервісів, що є “потенційно небезпечними” для користувачів – за дослідженням компанії “Positive Technologies” [1]. Дві третини перевірених ними сайтів виявились такими, що, за їх словами, мають критичні уразливості.

Однією з найпоширеніших вразливостей веб-додатків є міжсайтовий скриптинг (англ. “Cross-Site Scripting” або XSS). XSS атаки направлені на впровадження чужорідного для системи коду з метою отримання доступу до персональних даних користувачів або конфіденційної інформації у системі [2].

XSS виникає, коли на сторінки, які були згенеровані сервером, з якоїсь причини потрапляють користувацькі скрипти (сценарії). Нижче наведено невеликий список основних методів боротьби з міжсайтовим скриптингом, рекомендовані “Open Web Application Security Project (OWASP)” [3,4]:

- валідація всіх вхідних даних, наданих клієнтською стороною, на стороні серверу. Розробник має програмно передбачати ймовірність потрапляння чужорідного коду у систему через надані текстові дані зі сторони клієнту. Передбачаючи, які дані необхідно ввести користувачу і зменшуючи область допустимих значень (використовуючи, як приклад, відповідні значення аргументу “type” HTML-тегу “input” для різних типів даних), можна вже на етапі валідації даних (як на стороні клієнту, так і на стороні серверу) захистити власний веб-застосунок від XSS атак;

- екранування всіх вхідних даних. Навіть якщо код зловмисників потрапив на сервер, екранування тексту при його вбудовуванні в HTML-сторінку захистить систему від виконання небажаного коду;

- використання HTTP заголовку “X-Content-Security-Policy” (з англ. “Політика захисту контенту”). За допомогою цього заголовку можна вказати браузеру, які джерела коду (а також стилей, шрифтів, медіафайлів тощо) є надійними, а які є небезпечними;

- вказання кодування для всіх полів вводу на сторінці, наприклад, ISO-8859-1 або UTF-8;

– забезпечення захисту передачі і збереження Cookies, що реалізується за допомогою обмеження допустимих доменів для передачі та прийому Cookies, використання параметру HttpOnly, TLS та інш.

– визначення HTTP заголовку “Access-Control-Allow-Origin”;

– використання названих вище порад може захистити розроблюваний вами веб-додаток від XSS атак, але все одно не дає стовідсоткової гарантії, оскільки результат залежить від того, наскільки вдалими є методи валідації, наскільки багато можливостей потрапляння небезпечних даних на сервер передбачає сам розробник.

Для найбільш ефективного захисту Вашого додатку, особисто я рекомендую використовувати найактуальніші методології і інструмент для розробки веб-застосунків: відмовитися від генерації сторінок на стороні серверу та реалізовувати взаємодії клієнтської та серверної частини через програмний інтерфейс (API), проводити екранування усіх текстових даних, що потрапляють на сервер, надавати користувачеві альтернативні варіанти розмітки та форматування тексту, відмінні від HTML (наприклад, BB-code, Markdown тощо), використовувати фреймворки для розробки, що мають захист від XSS атак за замовчуванням (наприклад, Django, Ruby on Rails, ASP.NET і інші), а також використовувати HTTP заголовок відповіді “X-Content-Security-Policy”, що наразі підтримується більшістю сучасних веб-браузерів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Міжсайтовий скриптинг: Вікіпедія [Електронний режим] – Режим доступу:

https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B6%D1%81%D0%B0%D0%B9%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%81%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%B8%D0%BD%D0%B3

2. Веб-вразливості. Неймовірне – очевидно: Хабр [Електронний ресурс] – Режим доступу: <https://habr.com/ru/company/pt/blog/145329/>

3. Безпека в PHP (частина 3): Хабр [Електронний ресурс] – Режим доступу: <https://habr.com/ru/company/mailru/blog/352442/>

4. A7-Cross-Site Scripting (XSS): OWASP [Електронний ресурс]. – Режим доступу: [https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A7-Cross-Site_Scripting_\(XSS\)](https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A7-Cross-Site_Scripting_(XSS))

ЗАХИСТ ВІД НЕСАНКЦІОНОВАНОГО КОПІЮВАННЯ

Під системою захисту від несанкціонованого використання і копіювання розуміється комплекс програмних або програмно-апаратних засобів, призначених для ускладнення або заборони нелегального поширення, використання і (або) зміни програмних продуктів та інших інформаційних ресурсів. Термін «нелегальне» тут розуміється як вироблене без згоди правовласника. Нелегальне зміна інформаційного ресурсу може знадобитися порушнику для того, щоб змінений їм продукт не підпадав під дію законодавства про захист авторських прав.

Також очевидно, що надійність системи захисту дорівнює надійності найменш захищеного з її модулів.

Виділимо принципи створення і використання систем захисту від копіювання:

- умови поширення програмних продуктів;
- обрахунок можливостей користувачів програмного продукту по зняттю з нього системи захисту;
- властивості розповсюджуваного програмного продукту;
- оцінка можливих втрат при знятті захисту і нелегальному використанні;
- особливості рівня знань і кваліфікації осіб, що знімають систему захисту;
- постійне оновлення використаних в системі захисту коштів.

Виділимо основні компоненти системи захисту програмних продуктів від несанкціонованого копіювання:

- модуль перевірки ключової інформації – може бути доданий до здійснено коду захищається програми за технологією комп'ютерного вірусу [1];
- блок захисту від вивчення алгоритму роботи системи захисту;
- модуль відповідної реакції в разі спроби несанкціонованого використання.

Найбільш дієвим (після криптографічного перетворення) методом протидії несанкціонованому виконання скопійованих програм є використання блоку контролю середовища розміщення програми.

Приклад застарілого методу захисту. Деякі старі програми для DOS створювали некопійовані мітки на жорсткому диску [2]. Наприклад, файл

довжиною 1 байт займає на диску один кластер (не менше 512 байт), і в що залишилися 511 байт можна записати деяку інформацію. Зараз ця практика не використовується, так як вона чревата втратами даних.

Проаналізувавши метод захист від несанкціонованого копіювання була виявлена поширена проблема, система захисту від копіювання неминуче створює користувачеві незручності, і тому, з точки зору користувача, копія, з якою захист знята, навіть якщо і незаконно, в якомусь сенсі краще ліцензійної.

Наприклад:

- з жорсткого диска дані завантажуються швидше, ніж з зовнішнього носія;
- захищену музику, завантажену на портативний плеєр, неможливо прослухати, наприклад з домашнього ПК;
- апаратні технології захисту призводять до подорожчання пристроїв.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Фролов А.В. MS-DOS для програмістів / А.В. Фролов, Г.В. Фролов.– Диалог-МИФИ, 1995. – 253с.
2. Зубков С.В. Assembler для DOS, Windows і UNIX / С.В. Зубков, Н. А. Космічева. – ДМК Пресс-Москва, 2000. – 602 с.

УДК 004.5

Архипов А.С.¹, Туленков А.В.², Пархоменко А.В.³

¹ студ. гр. КНТ-126 НУ «Запорізька політехніка»

² асп. НУ «Запорізька політехніка»

³ канд. техн. наук, доц. НУ «Запорізька політехніка»

РОЗРОБКА ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ ДЛЯ КОНТРОЛЮ ПОКАЗНИКІВ ЕКСПЛУАТАЦІЇ ОБЛАДНАННЯ ДОМАШНЬОЇ АВТОМАТИЗАЦІЇ

Сьогодні електрошитове обладнання є невід'ємною складовою будь якої системи домашньої або промислової автоматизації. Зважаючи на те, що за статистикою в Україні щороку від пожеж в електричних щитах гине приблизно 500 осіб, безпечна експлуатація електричних щитів є складною проблемою. Саме тому, важливо вдосконалювати апаратні та програмні засоби захисту та оповіщення при надзвичайних ситуаціях, які можуть стати причиною виходу з ладу обладнання та призвести до людських втрат.

Як показали проведені дослідження, існують декілька стандартів, які описують способи захисту оболонки технічного пристрою від попадання

зовнішніх твердих предметів та (або) вологи (IEC 60529, DIN 40050, ГОСТ 14254) та застосовуються для електричних щитів. Щити, які виготовлені за цими стандартами, встановлюються в приміщеннях з агресивними умовами навколишнього середовища, а також зовні приміщень. Також існують стійкі до високої температури кабельні вироби. Найкращий спосіб забезпечити максимальну безпеку при експлуатації електрощитового обладнання – це забезпечити високі вимоги до зовнішнього корпусу щита з термостійким обладнанням всередині. Але такий підхід спричиняє надто високу вартість обладнання. До того ж, є багато вже реалізованих раніше проєктів, для яких заміна обладнання є неможливою або недоцільною з фінансової точки зору. Тому, актуальною задачею є створення бюджетного програмно-апаратного комплексу для контролю показників при експлуатації щитового обладнання, який можливо встановити у щити, що експлуатуються та використовувати в нових проєктах як додатковий спосіб забезпечення надійної експлуатації.

Метою роботи є створення прототипу програмно-апаратного комплексу для контролю температурних показників та показників вологості, що буде встановлюватися в електричний щит та забезпечить оперативне сповіщення власникам або фахівцям про відхилення від нормальних умов експлуатації та відключення обладнання.

Як показав аналіз компонентів, які можуть бути застосовані при реалізації проєкту, важливими складовими можуть стати електронні реле. Наприклад, реле TER-9 застосовуються для комплексного контролю температури рідин і повітря в опалювальних або охолоджувальних системах, розподільних щитах, двигунах за допомогою виносних датчиків. Серед переваг можливо зазначити наявність таймеру реального часу з денною і тижневою програмами, наявність вбудованої пам'яті для збереження часто використовуваних температур, LCD дисплей з підсвічуванням, великий вибір функцій і контрольованих зон. Терморегулятор TER-3 забезпечує можливість встановлення в електрощит, температурний контроль у промисловому обладнанні, наявний контроль несправностей датчика, світлодіодну індикацію стану роботи. Недоліком розглянутих рішень є те, що вони не забезпечують інформування власника або фахівців у разі надзвичайної ситуації та не відстежують показник вологості у щиті.

За результатами проведеного дослідження, в роботі пропонується модифікувати наявні засоби температурного контролю в електрощитовому обладнанні від компанії Legrand шляхом встановлення датчика температури та вологості, а також GSM/GPRS модуля. Датчик вологи забезпечить контроль обладнання та допоможе запобігти утворенню корозії та коротких замкнень. Завдяки GSM/GPRS модулю у разі виникнення надзвичайної ситуації можливо швидко сповістити власника або фахівців про небезпеку у електрощиті шляхом відправки СМС повідомлення. Розроблений прототип

програмно-апаратного комплексу для контролю показників роботи електрощитового обладнання розроблено на платформі Arduino з використанням датчика температури та вологості DHT 11, реле Songle SRD-05VDC-SL-C, GSM/GPRS модуля Sim 900a. Програмну частину реалізовано в середовищі розробки Arduino IDE. Завдяки GPRS можливо передавати показники у хмарний сервіс для зберігання та обробки. Подальший аналіз отриманих даних дозволить сформулювати рекомендації щодо оптимальної роботи обладнання системи домашньої автоматизації. Наступним кроком роботи є інтеграція розробки з системою MyHome від компанії Legrand та тестування на реальному об'єкті.

Використання розробки в навчальному процесі дозволить отримати знання та практичні навички студентам, що вивчають системи домашньої та промислової автоматизації, зокрема системи нотифікації та електрощитове обладнання. В майбутньому, компоненти прототипу можна замінити іншими для підвищення показників компактності та надійності з метою подальшої реалізації реального промислового зразка.

УДК 004.5

Задоян М.К.¹, Соколянський О.В.², Пархоменко А.В.³

¹ студ. гр. КНТ-116 НУ «Запорізька політехніка»

² асп. НУ «Запорізька політехніка»

³ канд. техн. наук, доц. НУ «Запорізька політехніка»

РОЗРОБКА МОБІЛЬНОГО ІНТЕРФЕЙСУ ДЛЯ ВІДДАЛЕНОГО КЕРУВАННЯ ГІБРИДНОЮ ЛАБОРАТОРІЄЮ

Як відомо, віддалені та віртуальні лабораторії надають багато переваг для студентів, оскільки вони доступні без обмежень через Інтернет. Гібридна лабораторія (ГЛ) «SmartHouse&IoT», що створена командою розробників кафедри програмних засобів, дає можливість вивчати принципи домашньої автоматизації онлайн. Ця лабораторія являє собою спеціальну платформу яка дозволяє вивчати принципи та структуру систем типу Розумний будинок. Зазвичай користувачі через спеціальний графічний інтерфейс веб-додатку в режимі реального часу взаємодіють з реальним обладнанням лабораторії, зображення якого транслюється за допомогою веб-камери. Іншим варіантом є взаємодія з віртуальною моделлю «розумного будинку». В той же час, дослідження та реалізація сучасних методів взаємодії користувачів з програмним та апаратним забезпеченням ГЛ на основі передових мобільних технологій є актуальним завданням.

Сучасні мобільні месенджери стали повноцінними комунікаційними

центрами, оскільки, окрім відправки повідомлень, надають можливість обмінюватися файлами, голосовими повідомленнями та проводити відео конференції. Дуже велика кількість людей різного віку користуються ними для комунікацій кожного дня. Саме тому, перспективним напрямком є розробка функцій керування ГЛ в середовищі такого застосунку.

На сьогоднішній день майже усі сучасні месенджери мають спеціальну технологію, що зветься чат-ботами. Чат-ботом називають програму, яка використовує для роботи інтерфейс месенджера, що дозволяє йому розпізнавати запити користувача та коректно реагувати на них. Будь-який чат-бот можна налаштувати під потреби користувача або сферу діяльності. Головною перевагою чат-ботів є відсутність потреби встановлення додаткових застосунків та реєстрації, а їх функціонування відбувається за допомогою популярних та знайомих месенджерів.

На сьогодні в мережі досить багато чат-ботів, які відрізняються між собою формою взаємодії з користувачем, призначенням, якістю реалізації, тощо. Існують кнопкові, текстові, inline-боти, комунікаційні чат-боти, маркетингові, функціональні та багато інших. Підтримку роботи чат-ботів заявили такі месенджери як Facebook Messenger, Viber, Telegram, Skype, Sender, Wechat, тощо. А оскільки всі месенджери розроблені на різних платформах та програмний інтерфейс (API) у них різний, то для кожного потрібно створювати персонального бота. Згідно статистики від квітня 2020 року Android користувачі віддають перше місце Viber а друге Telegram [1]. А згідно статистики від цієї ж дати користувачі систем Apple віддають перевагу Telegram, а на другому місці Viber [2]. Тож можемо зробити висновок, що Telegram та Viber-це найпопулярніші месенджери, але функціонал Viber дуже обмежений, та за більшою його частиною треба звертатися до розробників. Саме тому було обрано Telegram в якості месенджера, для якого буде розроблюватися чат-бот для ГЛ.

В результаті цієї роботи створено спеціальний інтерфейс для взаємодії з ГЛ. Розроблено детальний список команд, згрупованих за вмістом для зручного керування. Завдяки цим командам користувач зможе отримати інформацію про певний експеримент або про всі експерименти, що доступні для виконання, зайняти чергу для експерименту, а також надіслати контрольну дію до конкретного експерименту. Організація черг для експериментів та система авторизації користувача реалізуються на основі взаємодії з API лабораторії. Таким чином, у разі зміни логіки авторизації або роботи з чергами, чат-бот лабораторії автоматично міститиме останню версію реалізації без необхідності оновлення програмного коду. Обробка запитів API месенджера Telegram базується на технології Webhook. Ця технологія дозволяє підтримувати велику кількість користувачів, що, в свою чергу, забезпечує можливість автоматично зберігати дані користувачів під час

їх авторизації у базі даних та керувати ними.

Впровадження сучасного мобільного інтерфейсу для онлайн-лабораторії дозволить підвищити інтерес студентів до використання подібних освітніх ресурсів, особливо в нестандартних ситуаціях (наприклад, у період карантину).

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. The Most Popular Communication AndroidApps in UA according to GooglePlay [Електронний ресурс] – Режим доступу: <https://www.similarweb.com/apps/top/google/store-rank/ua/communication/top-free>

2. TheMostPopulariPhoneSocialNetworkingApps in UA accordingtoAppStore [Електронний ресурс] – Режим доступу: <https://www.similarweb.com/apps/top/apple/store-rank/ua/social-networking/top-free/iphone>

УДК 616-71

Гозлов В.В.¹, Туленков А.В.², Пархоменко А.В.³

¹ студ. гр. КНТ-137 НУ «Запорізька політехніка»

² аспірант НУ «Запорізька політехніка»

³ канд. техн. наук, доц. НУ «Запорізька політехніка»

ВИКОРИСТАННЯ ДАТЧИКА MAX30102 ДЛЯ ПРОЄКТУВАННЯ ПАРАМЕДИЧНОГО БРАСЛЕТУ

В сучасних системах моніторингу здоров'я людини особливе місце займають датчики контролю серцевої діяльності та насичення артеріальної крові киснем, які відрізняються компактністю та точністю. Це дозволяє вбудовувати їх в носимі пристрої, такі як смарт-годинники та фітнес-трекери.

Як показали проведені дослідження, серед датчиків, що відповідають таким критеріям, можна виділити оптичний датчик пульсоксиметрії MAX30102 розроблений компанією Maxim Integrated [1]. MAX30102 – це плата з інтегрованим сенсорним модулем, яка дозволяє відстежувати частоту серцевих скорочень (пульс), рівень насичення артеріальної крові киснем (SpO2) та значення температури навколишнього середовища. В плату інтегровано червоний та інфрачервоний світлодіоди, фотоприймач та оптичні елементи, а взаємодія із зовнішніми пристроями відбувається за допомогою послідовної шини даних ІС (I2C).



Рисунок 1 – Зовнішній вигляд плат MAX30102

Для визначення частоти серцевих скорочень за допомогою оптичного датчика MAX30102 використовують фотоплетізмограму – метод реєстрації кров'яного потоку за допомогою використання інфрачервоного або світлового випромінювання та фоторезистора [2].

Оптичний датчик пульсу за допомогою інфрачервоного світлодіода просвічує шкірний покрив та вимірює кількість відбитого світла на фоторезисторі. Чим більший кров'яний потік, тим менше світла поглинається в тканинах організму.

Після вимірів виділяють окремі періоди і рахують число піків на отриманому сигналі або знаходять час між двома скороченнями та в результаті розраховують значення пульсу.

Для визначення вмісту кисню в крові використовують пульсоксиметрію – простий неінвазивний (без пошкодження шкірного покриву) метод моніторингу насичення гемоглобіну артеріальної крові киснем [3]. Цей метод заснований на тому, що поглинання гемоглобіном світла двох різних по довжині хвиль змінюється в залежності від насичення його киснем, а світловий сигнал, проходячи тканини, набуває пульсуючий характер внаслідок зміни обсягу кровотоку при кожному серцевому скороченні.

Датчик пульсоксиметрії за допомогою червоного та інфрачервоного світлодіода просвічує шкірний покрив та вимірює кількість відбитого світла, що потрапляє на фоторезистор. З метою калібрування температурної залежності також вимірюється значення температури навколишнього середовища. На основі ряду отриманих результатів розраховується насичення артеріальної крові киснем, як відношення насиченого киснем гемоглобіну до загального вмісту гемоглобіну у відсотках.

Оптичний датчик MAX30102 вимогливий до щільного прилягання до шкіри, адже при потраплянні на нього інших джерел світла відбувається засвічування, а при зчитуванні даних з'являються шуми. Оскільки його робота полягає у визначенні кровотоку в тканинах, то чим більше даних, тим краще.

За рахунок мініатюрних розмірів, доступності та простоти використання, високої точності й швидкодії оптичний датчик MAX30102 є незамінним модулем для побудови розумних носимих пристроїв з системами моніторингу стану організму людини, зокрема парамедичного браслету для контролю стану пацієнтів в надзвичайних умовах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Maxim Integrated Products [Electronic resource]. – Access mode: <https://www.maximintegrated.com/en.html>
2. Photoplethysmogram [Electronic resource]. – Access mode: <https://en.wikipedia.org/wiki/Photoplethysmogram>
3. Pulse oximetry [Electronic resource]. – Access mode: https://en.wikipedia.org/wiki/Pulse_oximetry

УДК 004.891

Куш А.В.¹, Пархоменко А.В.², Гладкова О.М.²

¹ студ. гр. КНТ-219м НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

ОЦІНКА ЯКОСТІ ВІДЕО-КОНТЕНТУ ДЛЯ МЕДИЧНОЇ ГАЛУЗІ НА ПЛАТФОРМІ YOUTUBE

Як відомо, веб-сервіс YouTube надає користувачам широкі можливості та потужний функціонал для самовираження, розвитку чи заробітку. YouTube активно використовують в навчальних цілях, оскільки на ньому наочно представлена інформація практично з будь-якої теми.

Головною проблемою веб-сервісу є наявність значної кількості неякісного контенту, оскільки на сьогодні платформа не передбачає повноцінної перевірки контенту, що публікується. Тому, підбирати відео повинен експерт, а це потребує значних часових витрат.

На жаль, ця проблема є досить масштабною та особливо гостро стоїть саме в сфері медицини, адже неякісний контент при навчанні медичних працівників може коштувати комусь життя. Це підтверджують результати, наведені в [1], де зазначено, що під час перевірки наявних на каналі відео, пов'язаних з процедурою імобілізації перелому дистального радіуса (фіксація після перелому зап'ястя) з 68000 відео тільки 16 відповідали поставленим критеріям якості.

Як показали проведені дослідження, на сьогоднішній день немає чіткого визначення терміну «якість контенту», вимоги до якого залежать від ресурсу розміщення та виду контенту, а також алгоритмів, що забезпечать оцінку

якості контенту. Тим не менш, можна виділити п'ять спільних критеріїв якості для будь-якого виду контенту: користь (повинен нести практичну цінність); тематичність (зберігає єдину тематику); довіра (матеріал підкріплений посиланнями на використану літературу); унікальність та оптимізація контенту під ключові запити; актуальність; віральність (здатність поширюватись самостійно).

Таким чином, розробка рекомендаційної системи, яка дозволить сформулювати рекомендації користувачам YouTube щодо вибору певного відео з медичною інформацією за змістом, якістю, точністю та достовірністю, є актуальною науковою задачею.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Assessment of “YouTube” Content for Distal Radius Fracture Immobilization [Електронний ресурс] – Режим доступу: <https://www.sciencedirect.com/science/article/abs/pii/S193172041>

УДК 004.9

Малюков М.К.¹, Пархоменко А.В.²

¹ студ. гр. КНТ-116 НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

ВПРОВАДЖЕННЯ ТЕХНОЛОГІЙ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ ДЛЯ ПОПУЛЯРИЗАЦІЇ СУЧАСНИХ НАВЧАЛЬНИХ ЛАБОРАТОРІЙ

Розвиток технологій віртуальної та доповненої реальності є одним з ключових аспектів, що виділяється експертами на сучасному ринку. Чимало аналітичних компаній прогнозують популяризацію технологій віртуальної реальності в усіх сферах діяльності людини. Наприклад, згідно з даними Goldman Sachs, сукупний об'єм ринку програмного забезпечення для віртуальної та доповненої реальності у 2025 році складе 35 млрд. доларів, а сукупна аудиторія – 315 млн. користувачів (у 2017 році об'єм ринку складав приблизно 9,1 млрд. доларів [1]).

Використання технологій віртуальної реальності для популяризації сучасних навчальних лабораторій дозволить підвищити зацікавленість абітурієнтів та студентів, а також мотивувати їх на вивчення та реалізацію цих технологій.

Як показали проведені дослідження, для розробки віртуального туру по навчальній лабораторії доцільно використовувати сучасний фреймворк A-Frame на основі існуючого three.js. A-Frame це веб-фреймворк з відкритим кодом для створення віртуальної реальності [2]. В першу чергу, перевага

цього фреймворку полягає у сумісності з більшістю веб-бібліотек та фреймворків, що існують на даний момент. По-друге, при роботі з ним можна налаштовувати сцену лише в одному рядку HTML-коду для обробки світла, налаштувань WebVR, рендерінгу, тощо. По-третє, фреймворк забезпечує можливість роботи над крос-платформними проектами. Цей інструментарій дозволить розробити надійне програмне забезпечення для створення віртуального туру по лабораторії вбудованих систем та віддаленої інженерії кафедри програмних засобів.

Віртуалізація навчання є сучасним трендом, який доцільно розвивати задля популяризації освіти та науки.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Иванова А.В. Технологии виртуальной и дополненной реальности: возможности и препятствия применения. [Електронний ресурс]. - Режим доступу: <https://cyberleninka.ru/article/n/tehnologii-virtualnoy-i-dopolnennoy-realnosti-vozmozhnosti-i-prepyatstviya-primeneniya>.

2. A-Frame [Електронний ресурс]. – Режим доступу: <https://aframe.io/>.

УДК 004.522

Яремченко Є.В.¹, Туленков А.В.², Пархоменко А.В.³

¹ студ. гр. КНТ-116 НУ «Запорізька політехніка»

² асп. НУ «Запорізька політехніка»

³ канд. техн. наук, доц. НУ «Запорізька політехніка»

РОЗРОБКА ГОЛОСОВОГО СПОВІЩЕННЯ СИСТЕМИ ДОМАШНЬОЇ АВТОМАТИЗАЦІЇ ДЛЯ КОРИСТУВАЧІВ З ОСОБЛИВИМИ ПОТРЕБАМИ

Сьогодні автоматизація, централізація та інтелектуалізація домашнього простору є актуальними технологіями, що дозволяють створити більш комфортні та безпечні умови проживання людей з особливими потребами.

Для підвищення ефективності контролю системи MyHome від компанії Legrand, авторами раніше було розроблено мобільний застосунок MyHome Logger, що дозволяє вести журнал подій, які відбуваються в системі. Однак, з самого початку не планувалося, що користувачами розробленого програмного забезпечення можуть бути люди з особливими потребами. Наприклад, люди з вадами зору, які не сприймають графічну інформацію, або люди з вадами опорно-рухової системи, для яких використання смартфона є ускладненим або взагалі неможливим. Тому, метою даної розробки є розширення функціоналу мобільного застосунку MyHome Logger шляхом

реалізації голосового сповіщення, що дозволить повідомляти користувача про настання певних подій, які були обрані в налаштуваннях програми.

Як показали проведені дослідження, технології озвучування тексту, елементів інтерфейсу та іншої інформації вже присутні в операційних системах (ОС) Windows, MacOS, Android, iOS та ін. Наприклад, в Windows така система називається Екранним диктором і входить до складу ОС починаючи з Windows 2000. Екранний диктор спрощує використання комп'ютера для людей з порушеннями зору, дозволяючи виконувати типові завдання без використання монітора і миші. Додаток дозволяє зчитувати елементи, які відображаються на екрані, такі як кнопки і текст, і вступати з ними у взаємодію. За допомогою екранного диктора користувач може читати і писати повідомлення електронної пошти, переглядати Інтернет-сторінки і працювати з документами.

В Android також існує вбудований додаток для людей з вадами зору – Talkback. Основні функції програми: озвучування тексту або перетворення мови в текст; озвучування всіх клавіш, на які натискає користувач; озвучування інформації про вхідний дзвінок; читання файлів на пристрої; читання та озвучування тексту в браузері; створення звіту про дії на пристрої та ін.

Огляд існуючих text-to-speech (TTS) бібліотек для React Native показав, що наразі існує декілька варіантів реалізації. Непоганим рішенням є Expo-speech SDK, проте її використання в даному проєкті не є бажаним, оскільки MyHome Logger було розроблено на React Native з використанням коду нативних мов програмування Java для Android та Objective-C для iOS, а для використання модулів React Native Expo потрібно встановлювати додаткові бібліотеки, що призведе до збільшення розміру виконуваного файлу програми. Також не є бажаним використання Google Cloud Speech To Text, оскільки це платний онлайн сервіс, а в процесі роботи MyHome Logger може бути доволі багато подій, для яких потрібно згенерувати голосове сповіщення, що призведе до надсилання великої кількості запитів до сервісу і також до певних затримок в часі. Тому, для реалізації системи голосового сповіщення в MyHome Logger доцільно обрати бібліотеку React Native TTS. Обрана бібліотека має багато налаштувань для зручності користування нею. Наприклад, налаштування швидкості вимови, голосу озвучування, якості звучання та ін. Також важливим є те, що розробники постійно її покращують та адаптують для нових версій ОС Android та iOS.

Для впровадження даної програми в MyHome Logger потрібно додати екран, на якому можливо буде обрати події системи, які потрібно озвучити. Крім цього повинні бути передбачені елементи інтерфейсу для налаштування параметрів озвучування. Ці налаштування будуть збережені і тоді при надходженні подій від системи MyHome програма буде перевіряти тип події

та озвучувати її, якщо користувач увімкнув озвучування для цього типу подій.

Таким чином, впровадження даної програми в мобільний застосунок MyHome Logger дозволить зробити його придатним для використання людьми з особливими потребами. Проте, можливо відмітити і користь даної функції для інженерів, які встановлюють та налаштовують системи домашньої автоматизації MyHome, оскільки бувають випадки, коли одночасно від системи надходить багато подій і за ними важко стежити. Тоді інженер зможе обрати типи подій, що задіяні під час налаштування певних елементів системи, і отримувати голосове сповіщення про їх надходження.

Подальша інтеграція MyHome Logger з голосовим сповіщенням в систему MyHome дозволить різним категоріям користувачів ефективно та безпечно користуватися системою домашньої автоматизації, контролювати події та реалізовувати оптимальні сценарії поведінки.

УДК 004.946

Білов О.І.¹, Залюбовський Я.І.², Пархоменко А.В.³, Гладкова О.М.³

¹ студ. гр. КНТ-219м НУ «Запорізька політехніка»

² асп. НУ «Запорізька політехніка»

³ канд. техн. наук, доц. НУ «Запорізька політехніка»

ВІРТУАЛЬНІ СЕРЕДОВИЩА ДЛЯ ДОСЛІДЖЕННЯ СИСТЕМ РОЗУМНИЙ БУДИНОК

Використання середовищ віртуальної та доповненої реальності розширює рамки досліджень у різних галузях, в тому числі в сфері розробки систем Розумний будинок (СРБ). Мета даної роботи - розробка спеціалізованого віртуального середовища для дослідження технологій домашньої автоматизації та демонстрації особливостей СРБ.

На основі аналізу існуючого досвіду було вирішено реалізувати два різні підходи – веб-орієнтований та апаратно-орієнтований. Перший підхід спрямований на розробку інтерактивного веб-орієнтованого середовища для розширення функціональності віддаленої лабораторії Smart House & IoT. Створене віртуальне середовище базується на платформі Web GL у поєднанні з HTML5 та JavaScript. Для його реалізації також використовувався ігровий рушій Unity 5 та середовище Autodesk 3ds Max 3D.

Другий підхід спрямований на розробку віртуального середовища, що дозволяє людині з шоломом віртуальної реальності Oculus Go отримати досвід роботи з СРБ як зовні, так і всередині будівлі та взаємодіяти з датчиками та приводами. Для реалізації цього підходу були використані мова

програмування C # та кросплатформне середовище розробки комп'ютерних ігор Unity3D.

Сьогодні розроблене веб-орієнтоване середовище доступне на веб-сайті віддаленої лабораторії Smart House & IoT та дозволяє дослідникам отримати корисну інформацію про компоненти СРБ та управління різними підсистемами. Середовище віртуальної реальності для шолома Oculus Go також стабільно працює, дає дослідникам корисний досвід роботи з технологіями та системами домашньої автоматизації.

Використання створених середовищ віртуальної реальності дозволяє дослідити структурні та функціональні особливості СРБ, моделювати поведінку мешканців з метою створення ефективних сценаріїв управління. Майбутня робота буде спрямована на реалізацію додаткової функціональності, зокрема на створення декількох персонажів (або декількох мешканців СРБ) для веб-орієнтованого віртуального середовища віддаленої лабораторії Smart House & IoT.

УДК 004.42

Захряпа Р.А.¹, Кривцун О.В.²

¹ студ. гр. КНТ-147сп НУ «Запорізька політехніка»,

² канд. техн. наук, доц. НУ «Запорізька політехніка»

ПРОГРАМНА РЕАЛІЗАЦІЯ ГРИ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ UNREAL ENGINE

В наш час комп'ютерна техніка розвивається надзвичайно швидкими темпами. Комп'ютери набули широкого використання у багатьох сферах діяльності людини. Їх використовують і для роботи, і для приємного дозвілля. Саме для розваг були створені відеоігри, які розвиваються не менш швидко, ніж інформаційні технології в цілому. Комп'ютерні ігри мають великий попит на сучасному ринку інформаційних технологій, а значить існує і «попит» на розробників, які їх створюють. Отже, створення відеоігор є актуальною проблемою нашого часу.

Якщо немає нагоди вийти на вулицю та відпочити від повсякденної рутини, роботи та інших справ, які вже могли набриднути, для цього – були створені комп'ютерні відеоігри, які надають можливості весело та просто провести час разом із друзями, граючи у відеогру та одночасно дізнатися, що нового у твоїх друзів або знайомих.

На даний момент існує безліч ігор, в яких можна провести свій час, але не всі вони задовольняють потреби кожного користувача, тому, користуючись нагодою, було прийняте рішення про розроблення цієї

відеогри. Однак більшість сучасних ігор коштують немало, дозволити їх собі може не кожна людина, що бажає провести свій час у грі.

Метою виконання випускної кваліфікаційної роботи бакалавра є програмна реалізація гри з використанням технології Unreal Engine.

Інтерфейс необхідно розробити максимально зручний – простий, з підказками та орієнтирами для того, щоб можна було швидко орієнтуватися та розуміти що відбувається у даний момент часу у відеогрі.

Меню повинно надавати лише необхідну інформацію про налаштування стосовно гри та серверу.

Також необхідно зробити акцент на початку – для того щоб зрозуміти що гра розпочалась та можна почати діяти та на кінці гри – для щоб можна було побачити результати (хто виграв, скільки та який гравець набрав очок та інше).

Головний комп'ютер – гравець, який створює гру і є сервером, що зможе оброблювати запити інших клієнтів та надавати можливості виконувати ті чи інші дії. Сервер має можливості по обмеженню кількості гравців на полі бою, що можна буде зробити на початку гри перед її створенням.

Програмний продукт для клієнта та сервера є єдиним, але сервером буде виступати та людина, що створила гру, та серверу необхідно буде оголосити свою IP-адресу для того, щоб інші гравці змогли приєднатися до гри.

Основні задачі системи: можливість створити гру, можливість приєднатися до вже існуючої гри, надання максимально зручних та простих меню налаштувань, підрахунок зароблених очок для кожного гравця, підрахунок результатів гри по її закінченню, можливість задавати обмеження на кількість гравців в одній грі, можливість зміни зброї під час гри, можливість поповнення боєприпасів до зброї, можливість поповнення здоров'я гравця за допомогою аптечок.

Програмний продукт написаний за допомогою об'єктно-орієнтовної мови програмування C++ з використанням середовища розробки для написання коду Visual Studio 2017 Community, візуальної системи програмування Blueprints, яку надає середовище розробки Unreal Engine Editor. Середовищем функціонування програмного продукту обрана операційна система Microsoft Windows 10.

Для того, щоб запустити гру, вам необхідно буде встановити її за допомогою спеціального файлу встановлення, який матиме розширення «.exe».

Для того, щоб надати простий та зручний інтерфейс, було створено наступний інтерфейс меню (рис. 1).

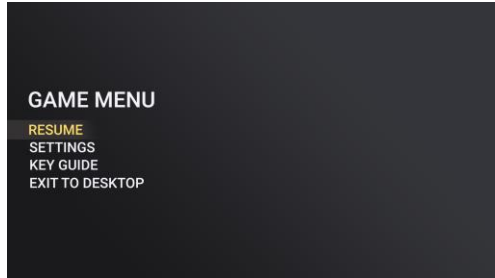


Рисунок 1 – Вигляд меню відеогри

Висновки

В роботі розглянута актуальність розробки відеогри для приємного дозвілля. Було прийнято рішення щодо реалізації гри з використанням технології Unreal Engine. На даний момент програмний продукт знаходиться у процесі розробки та буде готовий до використання найближчим часом.

УДК 004.422.81

Сердюк С.М.¹, Камінська Ж.К.², Шутко А.С.³

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² асист. НУ «Запорізька політехніка»

³ студ. гр. КНТ-128 НУ «Запорізька політехніка»

РОЗРОБКА ПРОГРАМИ НОТАТОК «ANOTES»

Використання нотаток – один зі способів встигати більше. Для деяких людей додаток для створення заміток – це не просто віртуальний блокнот з можливістю пошуку; це гарно організована система з безліччю складних функцій і опцій.

Існує багато додатків для Android, iOS, Windows та Mac OS, що роблять все ще легшим та зручнішим. У якості аналогів було розглянуто програми Simplenote та Evernote. У результаті проведеного аналізу було виявлено наступні недоліки, які притаманні Simplenote: програма не має можливості нагадати про майбутні події; не дозволяє пошук нотатки за назвою, або за фрагментом тексту. Недоліки Evernote: програма незграбна і складна; нові користувачі повинні інвестувати значну кількість часу, щоб вивчити її, перш ніж вони можуть фактично використати навіть половину потужності; вона дорого коштує.

У статті запропонована програма «ANotes», яка усуває деякі недоліки, які були вказані вище (має можливість нагадати про подію і дозволяє пошук

нотатки; програма не вимагає навчання, бо інтерфейс інтуїтивно зрозумілий.).

Для написання програмного продукту, було обрано середовище програмування Qt C++, так як дане середовище надає необхідний інструментарій для ефективного і швидкого створення кросплатформних застосунків і надає можливість зручно працювати з базами даних.

Розроблений програмний продукт, має функції створення, видалення, збереження, пошуку нотатки, функцію нагадування про подію. Також є випадające меню з функціями зміни email, зміни паролю, вихід з облікового запису (ОЗ), видалення ОЗ та виходу з програми, що розташовані у головному вікні (рис. 1а). Функції зміни email, зміни паролю мають однаковий інтерфейс.

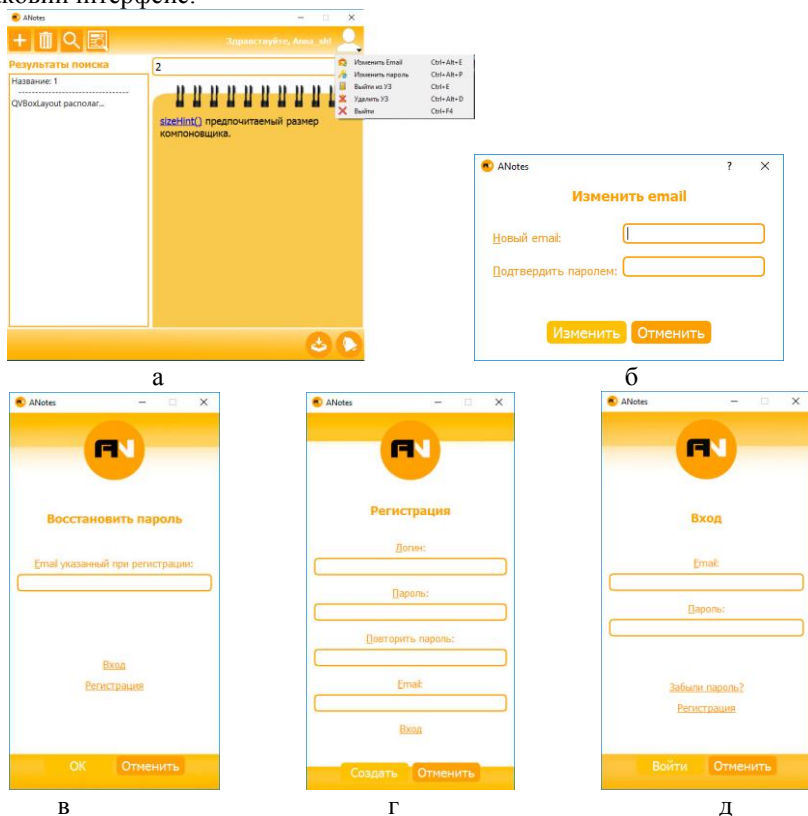


Рисунок 1 – Фрагменты интерфейсу програми

Приклад вікна розробленого інтерфейсу представлено на рис. 1б. При вході в програму треба створити ОЗ, або вийти в існуючий ОЗ. Якщо користувач забув пароль є функція відновлення пароллю. Приклад вікон реєстрації і входу представлено на рис. 1в і рис. 1г, а відновлення пароллю – на рис. 1д.

Таким чином, була розроблена програма «ANotes». Серед її переваг: зручний та простий інтерфейс дозволяє швидко розібратися в основних функціях і можливостях не тільки досвідченим користувачам, а і новачкам; доступність застосунок в будь-який час доби; застосунок повністю автономний.

УДК 004.5

Сердюк С.М.¹, Камінська Ж.К.², Заїкіна К.Є.³

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² асист. НУ «Запорізька політехніка»

³ студ. гр. КНТ-128 НУ « Запорізька політехніка »

РОЗРОБКА КОНЦЕПТУ «РОЗУМНА ШАФА»

Розумна шафа – це абсолютно нова категорія побутової техніки, відмінне рішення для речей, які вимагають особливого догляду. Дуже мало сучасних компаній, які представляли цей вид техніки на ринку України, в той час коли в Південній Америці стайлери для одягу у вигляді функціональних шаф набули широкого поширення.

У якості аналогів було розглянуто моделі LG Styler (компанія LG) – системи догляду за одягом за допомогою функції пари. Дві моделі, LG Styler ThinQ і LG Styler Mirrored Glass Door [1], оснащені додатковими функціями для зручності користування. Дані системи забезпечують антибактеріальну обробку речей, сушіння та мають функцію управління за допомогою голосових команд.

У результаті проведеного аналізу було виявлено наступні недоліки, які притаманні побутовій техніці LG Styler: мала кількість функцій, шумна робота, велике споживання електроенергії, довга тривалість режимів роботи функцій, висока вартість, недоступність для системи IOS.

У статті запропоновано концепт «Розумної шафи», який усуває деякі недоліки, які були вказані вище (тобто розумна шафа має більшу кількість функцій та доступна для будь-якого смартфона). Також розумну шафу можливо інтегрувати в систему розумного будинку.

Для написання програмного продукту, було обрано середовище програмування Visual Studio 2017 C#, так як дане середовище надає

необхідний інструментарій для ефективного і швидкого створення застосунків з графічним інтерфейсом.

Розроблений програмний продукт, має функції сушіння, прасування, дезінфекції та замовлення одягу через інтернет (функція замовлення одягу через інтернет здійснюється через онлайн-магазин Lamoda.ua.). У верхній частині головного вікна, яке представлено на рисунку 1 а), розташовано логотип розумної шафи. Далі користувачу пропонується перелік функцій, користувач може вибрати потрібну йому функцію. Функції сушіння, прасування та дезінфекція одягу мають однаковий інтерфейс. Приклад вікна цього інтерфейсу представлено на рисунку 1 б). У правому верхньому куті, позначено зображення нашої функції. Далі, пропонується список матеріалів тканини, користувач може вибрати одну з них, відповідно до свого одягу. Температура автоматично виставляється, після того як вказано матеріал тканини (користувач може знизити виставлену шафою температуру. Щоб не пошкодити одяг, користувач не може збільшувати температуру). Інформаційне вікно виведення залишеного часу розміщено знизу після вікна з температурою, час також виставляється автоматично. Кнопку запуску процедури, розміщено в нижній частині. Елемент управління, який здійснює вихід на головне вікно, розміщено в верхній частині інтерфейсу з вирівнюванням по лівій стороні.

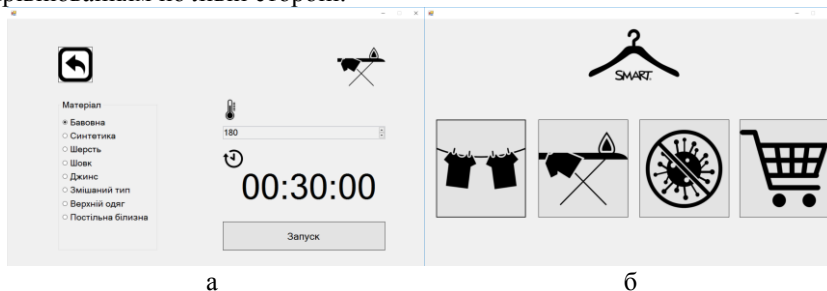


Рисунок 1 – Фрагменти інтерфейсу програми
а – головне вікно; б – вікно функції прасування одягу

Таким чином, було розроблено концепт «Розумної шафи». Серед його переваг: зручний та простий інтерфейс, можливість інтеграції в розумний будинок, наявність базових можливостей, доступність застосунку в будь-який час доби, з будь-якого смартфона.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. LG Styler ThinQ and LG Styler Mirrored Glass Door [Електронний ресурс]. – Режим доступу: <https://root-nation.com/news/partners-news/ru-lg-styler-ifa-2018/>

ІНТЕРФЕЙС СИСТЕМИ ЕЛЕКТРОННОГО МЕНЮ

З кожним днем потреба зручно та швидко зробити замовлення у закладі харчування зростає. Довготривале очікування офіціанта негативно позначається на настроях відвідувачів, залишаючи погані враження про заклад в цілому.

Також недоліками звичної системи здійснення замовлення є відсутність можливості:

- обрати зручну для відвідувача мову меню;
- ознайомитись з детальною інформацією про страву (наприклад, часом приготування);
- швидко знайти потрібну страву за категорією або назвою;
- за необхідності відредагувати замовлення (видалити страву із замовлення або змінити кількість порцій).

Вирішення цих проблем полягає у автоматизації процесу обслуговування шляхом розробки системи електронного меню. Аналогом системи, що розглядаються в даній статті, в Україні є термінали самообслуговування в досить масштабній мережі McDonalds. Використання подібної системи значно зменшило завантаженість касирів та прискорила обслуговування, результатом чого зменшення черг.

Отже, користувачами людино-машинного інтерфейсу (ЛМІ) є відвідувачі закладів харчування. Також користувачами можуть бути і офіціанти для виконання функції моніторингу стану страви(готова чи не готова) та кухарі для отримання замовлень та інформації щодо особистих побажань відвідувача. Ніяких обмежень для всіх типів користувачів немає.

В процесі аналізу системи «користувач – заклад харчування» була розроблена функціонально-часова та функціонально-об'єктна структури, що дозволило визначити сценарії дій користувачів у робочих, чи проблемних ситуаціях, та виявити номенклатуру елементів ЛМІ, що мають відображати необхідну інформацію.

Таким чином, був проведений системний аналіз та визначено основні функціональні вимоги до інтерфейсу ЛМІ, що проектувався (рис.1).

ЛМІ не передбачає повну заміну обов'язків офіціанта. Для певних потреб (наприклад, для розрахунку готівкою) відвідувач може викликати офіціанта, натиснувши відповідну кнопку.

Практична цінність розглянутого підходу та розробленого на його основі ЛМІ, полягає у скороченні часу на здійснення замовлення в закладах харчування, підвищенні рівня обслуговування в цих закладах, зменшенні психічного стресу та мінімізації ризику виникнення помилок через людський фактор.

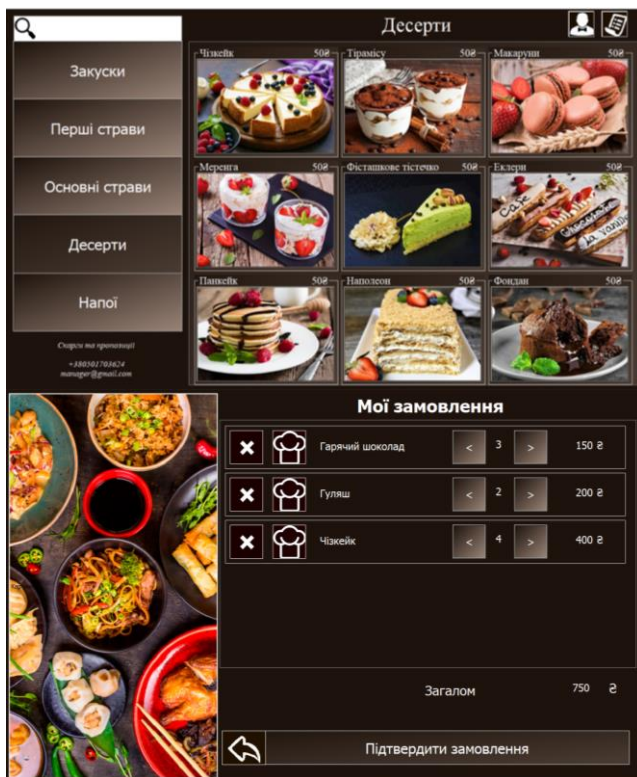


Рисунок 1 – Фрагменти інтерфейсу програми

ПРОБЛЕМНО-ОРІЄНТОВАНИЙ ІНТЕРФЕЙС БАГАТОФУНКЦІОНАЛЬНОГО БОРТОВОГО КОМП'ЮТЕРА

Під час дорожнього руху, найбільший ризик аварій створює людський фактор. Оскільки, крім керування транспортним засобом (ТС), водій має виконувати ще й інші ресурсоемні задачі (моніторинг маршруту, інформації про поточний технічний стан ТС, створення комфортних умов для себе і пасажирів, керування периферією тощо), увага водія до дорожньої обстановки падає, внаслідок чого можуть виникати критичні ситуації.

Для запобігання аварій і підвищення інформованості водіїв в сучасних автомобілях використовують інтелектуальні помічники у вигляді бортових комп'ютерів (БК).

Ціллю даного дослідження є розробка підходу до проблемно-орієнтованого проектування багатофункціонального інтерфейсу БК.

Всі БК, умовно, можуть бути розділені на два типи: вузькоспеціалізовані, що виконують конкретні функції (електронні системи керування, маршрутні комп'ютери, діагностичні системи, інформаційні електронні панелі) та універсальні, що об'єднують різні функції (інформаційні, діагностичні, розважальні, функції керування, інші).

Найбільш відомими БК є NAVISURFER II UBU-3G; Multitronics; ParkCity ELM-327WF та AutoPC. Недоліками цих БК є відсутність:

- відображення другорядних, але важливих даних про авто (запас ходу, рівень заряду акумулятора, поточний час);
- системи оповіщення на мобільний телефон;
- засобів керування периферією (швидкий доступ до налаштування або активації/деактивації елементів периферії).

Для формалізації знань про систему “водій-машина-середовище” було проведено компонентний та морфологічний аналіз. В процесі компонентного аналізу розроблена множина компонентно-системних, компонентно-елементних, компонентно-функціональних та компонентно-режимних структур, задля виявлення типу користувачів ЛМІ (водій), контексту використання інтерфейсу та потреб, які можуть виникати у процесі діяльності водія. В процесі морфологічного аналізу розроблена множина функціонально-часових та функціонально-об'єктних структур, що дозволило визначити сценарії дій користувачів та виявити номенклатуру елементів ЛМІ.

Таким чином, проведений системний аналіз, та аналіз аналогів визначили основні функціональні вимоги до універсального БК, та ЛМІ, що проектувався (рис.1).

Практична цінність розглянутого підходу та розробленого на його основі ЛМІ, полягає у підвищенні високорівневої ситуаційної обізнаності користувачів, ефективності їх діяльності, мінімізації зусиль і ризику помилок та зменшення психічного стресу.

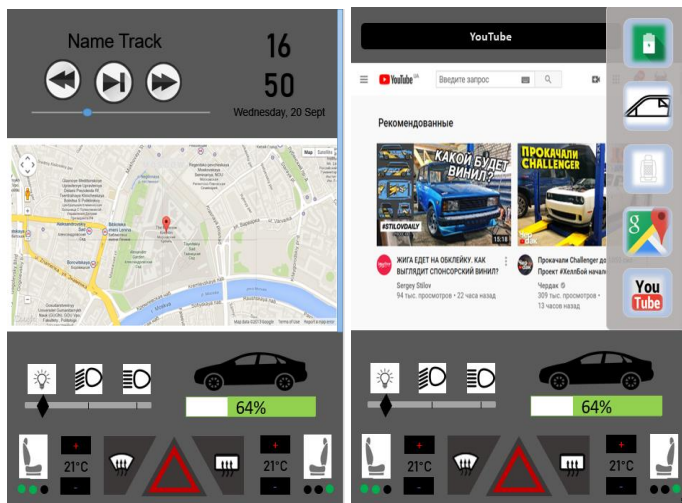


Рисунок 1 – Фрагменти інтерфейсу програми

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Муратов О.Э., Тихонов М.Н. Человеческий фактор на ядерных объектах [Электронный ресурс], 2008. – Режим доступа: <http://www.proatom.ru/modules.php?name=News&file=print&sid=1555>
2. Информационно-управляющие человеко-машинные системы: исследование, проектирование, испытание. Справочник / А.Н. Адаменко, А.Т. Ашеро́в, И.Л. Бердни́ков и др.; под общ. ред. А.И. Губинского. – М.: Машиностроение, 1993. – 528 с.

РОЗРОБКА ІНТЕРФЕЙСУ МУЗИЧНОГО ПЛЕЄРА ДЛЯ ГІТАРИСТІВ

У теперішньому часі активно розвивається музична індустрія. Все більше людей є зацікавленими не тільки у прослуховуванні, а й у відтворенні музики власноруч. Музичні школи не завжди успішно виконують задачі з навчання всіх бажаючих музиці та нотній грамоті. Тому для вирішення цієї проблеми були створені так звані архіви з інструментальними партіями для різноманітних композицій. На жаль, програмних продуктів у цій ніші дуже мало, і вони пропонують обмежений функціонал, а доступ до всіх можливостей програми можна отримати лише платно.

Так як існує декілька аналогів застосунку, що розглядається у даній статті, для структурного розбору музичних композицій, було розроблено інтерфейс музичного плеєра для гітаристів, який комбінує у собі традиційний функціонал для прослуховування музики та можливість вивчення композиції за табулатурою. Основними вимогами до створення інтерфейсу були:

- можливість обирати режим відтворення композиції (звичайне відтворення або у виді табулатури);
- можливість налаштування критеріїв відтворення композиції у режимі табулатури (регулювання швидкості відтворення композиції, відтворення партій окремих інструментів, можливість відтворення композиції без звукового супроводження, наявність метроному);
- можливість пошуку композиції за різними критеріями (жанр, альбом, назва, виконавець);
- можливість зупинки/перемотки композиції на будь-який її момент для повторного відтворення.

Розроблений підхід заснований на функціонально-структурної теорії ергатичних систем А.І. Губінського [1], що узагальнює стратифікований підхід до подання систем. Для формалізації знань про систему “користувач-середовище” було проведено її компонентний та морфологічний аналіз. В процесі морфологічного аналізу було розроблено функціонально-об’єктну структуру, що дозволило визначити сценарії для дії користувачів у робочих ситуаціях та виявити номенклатуру елементів людино-машинного інтерфейсу (ЛМІ), фрагменти якого наведені на рис.1.

В процесі компонентного аналізу було розроблено компонентно-функціональну структуру, що дозволило виявити: користувачів ЛМІ

(музиканти); контекст використання інтерфейсу; проблемні ситуації, які можуть виникнути у користувачів.

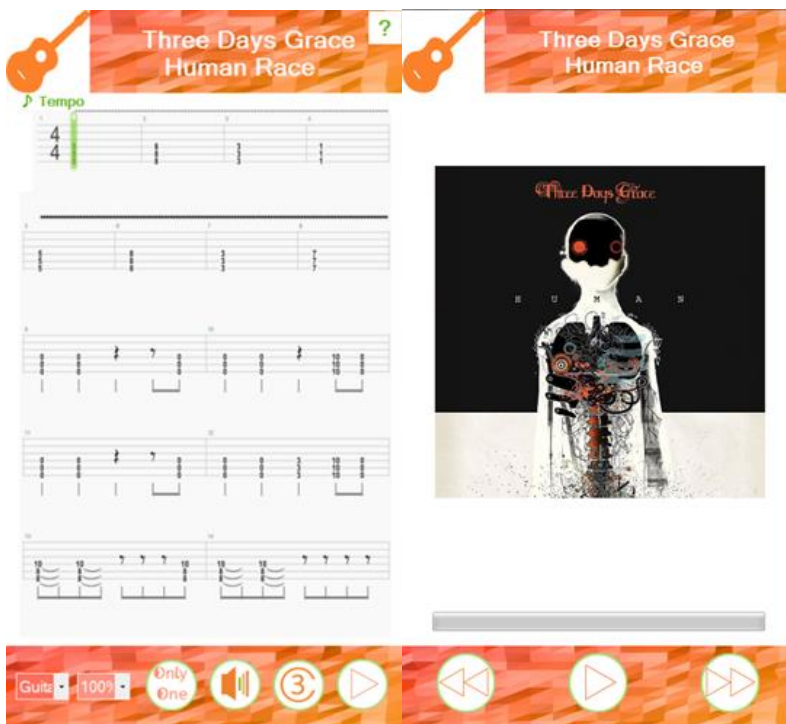


Рисунок 1 – Фрагменти інтерфейсу програми

Практична цінність підходу до розробки ЛМІ, полягає у досягненні ефективності, практичності, комфортності та доступності до архіву музичних композицій для їх подальшого вивчення або відтворення.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Информационно-управляющие человеко-машинные системы: исследование, проектирование, испытание. Справочник / А.Н. Адаменко, А.Т. Ашеро́в, И.Л. Бердни́ков и др.; под общ. ред. А.И. Губинского. – М.: Машиностроение, 1993. – 528 с.

ПРОБЛЕМНО-ОРІЄНТОВАНИЙ ІНТЕРФЕЙС РОЗУМНИХ ВАГІВ

Поки розумні ваги ще не увійшли в наше повсякденне життя, всі користуються звичайними вагами. Зараз предметна область розробки розумних ваг стрімко розвивається, але на даний момент розумні ваги мають багато недоліків. Розумні ваги можуть використовуватися для запобігання всіляких хвороб наприклад: ожиріння, остеопорозу і т.д. А також розумні ваги, допомагають дізнатися повну інформацію про склад тіла людини. Так як відсутня єдина, загальновізнана технологія проектування людино-машинних інтерфейсів (ЛІМІ), метою даного дослідження є розробка підходу до проблемно-орієнтованого проектування інтерфейсів даного типу.

На даний момент існує багато аналогів, таких як (Xiaomi Mi Smart Scale 2, Mirta SB-3125 SMART і т.д.), але вони мають багато недоліків таких як :

- швидкість роботи. Дуже велика кількість непотрібного, і навіть застарілого коду, що гальмує програму;
- висока ціна. Великі компанії прагнуть отримати більший прибуток, і через це ваги стають дорожчими;
- неоднозначність дизайну. Через велику кількість розробників можна побачити різний дизайн в різних вікнах.

Ваги розроблені на мові програмування C# яка вважається однією найшвидший мов програмування. Програма стабільно працює завдяки невеликій кількості функцій та добре оптимізованому коду.

Вимірювання показників тіла відбувається за допомогою біоімпедансного аналізу [1], за допомогою його реалізовані функції виміру: кількості жиру, води, вісцелярного жиру, м'язів, кісток в організмі. На даний момент біоімпеданс є найвідомішим та найдешевшим способом вимірювання показників складу тіла, але в нього є наступний недолік: не завжди вірно відображення показників через залежність електричного опору тіла людини від стану в якому вона знаходиться [2]. Також однією з функцій ваг є вимірювання індивідуальної маси тіла. (ІМТ), яка розраховується за формулою $ІМТ = \frac{ВАГА}{РІСТ^2}$ [3].

Завдяки тому що ваги не представляють якийсь відомий бренд вони не будуть мати великої націнки за товар, тому вони будуть досить дешевими у порівнянні з аналогами.

Ваги мають зрозумілий та легкий в освоєванні інтерфейс наведений на

рис. 1. Він має гострі кути, що відображають його строгість та цілеспрямованість. Для вибору функцій були використані кнопки, які розташовані в нижній частині дисплею. На кнопках є як текст так і умовні знаки показників які повинні відображатися. Завдяки використанню кодування умовними знаками вагами можуть користуватися люди, що не знають української мови, або які мають поганий зір.



Рисунок 1 – Фрагменти інтерфейсу програми

Практична цінність розглянутого підходу та розробленого на його основі ЛМІ, полягає у підвищенні обізнаності користувачів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Биоимпедансный анализ состава тела человек / Д. В. Николаев, А. В. Смирнов, И. Г. Бобринская, С.Г. Руднев. – Москва: Наука, 2009. – 392 с.
2. JA Máttar. Application of total body bioimpedance to the critically ill patient // New Horizons. – 1996. – Т. 4, № 4. – С. 493 – 503.
3. Body mass index and gastroesophageal reflux disease: a systematic review and meta-analysis/ Corley D. A., Kubo A. – 2006. – Т. 101. – №. 11. – P. 2619 – 2628.

УДК 004.5

Сердюк С.М.¹, Камінська Ж.К.², Терлецький С.В.³, Денисенко С.В.³,
Ліпихін С.Б.³

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² асист. НУ «Запорізька політехніка»

³ студ. гр. КНТ-138 НУ «Запорізька політехніка»

РОЗРОБКА КОНЦЕПТУ «РОЗУМНИЙ БУДИНОК»

В сучасному світі спостерігається безпрецедентне зростання кількості гаджетів та предметів так званого «Інтернету речей», одним з проявів якого є концепт високотехнологічного, інтелектуального, «розумного» будинку. Він дозволяє ефективно спростити та автоматизувати більшість завдань, які пов'язані з повсякденною, рутинною роботою у сфері догляду та підтримки контролю комфортабельності житла користувача.

Одна з перших реалізацій цієї концепції була втілена ще у 1966 році компанією Westinghouse Electric, а саме Джимом Сазерлендом, та мала назву «електронний домашній оператор ЕСНО IV» [1]. Однак на території України ця технологія не отримала широкого розповсюдження. Перші помітні результати почали з'являтися лише два роки тому: у грудні 2017 було заявлено про встановлення першої, розробленої в Україні, системи «розумний будинок» в Києві [2].

Отже, зважаючи на таку негативну тенденцію та недостатність розвитку цієї перспективної галузі, за мету цього дослідження було узято створення інтерфейсу для системи автоматизації квартири, актуальність якого полягає у адаптації для вітчизняного ринку, а саме – спрямованість на українського користувача.

В результаті проведеного аналізу інформації виявлено наступні проблеми та побажання користувачів:

- користувачі кожен день стикаються з побутовими проблемами, та недостатньою комфортабельністю управління житлом, а також бажають аби більшість рутинних справ, на кшталт керування світлом, теплом, водою, тощо, виконувалося автоматично;

- найважливішими справами для користувачів є проблеми, пов'язані із контролем електроенергії, тепла, води, охорони тощо.

Наступний крок у створенні продукту - це вибір цільового гаджету для управління системою. Керуючись принципами максимальної впізнаваності технології, а також керуючись досвідом аналогічного програмного забезпечення, можна дійти висновку, що планшет буде найкращим варіантом. Він є доволі зручним у використанні та відповідає наступним вимогам користувачів: розповсюдженість у повсякденному житті, достатні

розміри екрану і портативність.

Окрім того, однією із нагальних проблем для створення продукту є якість візуального оформлення і адаптивність, можливість підстроювання під широкий спектр пристроїв з різними технічними параметрами. Рішенням даної колізії є використання платформи Windows Presentation Foundation, у основі якої лежить векторна система візуалізації, яка не залежить від розширення екрану пристрою виведення і створена з урахуванням можливостей сучасного графічного обладнання [3].

У якості основи візуального представлення було використано шаблон «центральна сцена», який допомагає привернути погляд користувача до найбільш важливої інформації. Чітко виражений центральний елемент привертає увагу, відображаючи перехід між «підвінками» продукту, де головні кнопки керування винесені у статичну панель, а центр відводиться під основну інформацію, що відображається, прямокутна ж компоновка елементів якнайкраще гармонує з формою планшета (рис.1).

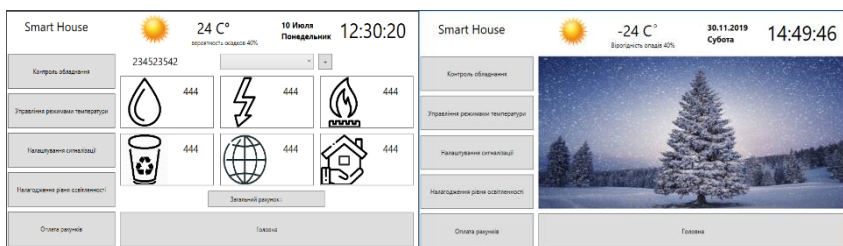


Рисунок 1 – Фрагменти інтерфейсу програми

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. If You Can't Stand the Coding, Stay Out of the Kitchen: Three Chapters in the History of Home Automation [Electronic resource]. – Access mode: <https://www.drdbobbs.com/architecture-and-design/if-you-cant-stand-the-coding-stay-out-of/184404040>
2. Система CLAP [Електрон. ресурс]. – Режим доступу: <https://clap.ua/>
3. Introduction to WPF in Visual Studio [Electronic resource]. – Access mode: <https://docs.microsoft.com/en-us/dotnet/framework/wpf/getting-started/introduction-to-wpf-in-vs?redirectedfrom=MSDN>

РОЗРОБКА ПРОБЛЕМНО – ОРІЄНТОВАНОГО ІНТЕРФЕЙСУ РОЗУМНОГО ХОЛОДИЛЬНИКА

Ми живемо в час розвитку інформаційних технологій, тож зараз дуже важко знайти таку техніку, яка не була би «розумною». Будь який прилад має свій розумний аналог в абсолютно кожній компанії, яка займає навіть маленьку долю на ринку побутової техніки.

Проте, головна проблема всієї розумної техніки це те, що окрім важливих функцій вона має і багато зайвих. І саме ці непотрібні функції потребують використання зайвих ресурсів, що позначається на ціні та комплектації.

У якості аналога було проаналізовано холодильник Samsung Family Hub [1], який окрім функції зберігання продуктів у холоді, має ще функцію доступу до інтернету (що дозволяє керувати ним за допомогою смартфона), сенсорну панель на якій можна малювати, дивитися фото та навіть програвати музику.

Для проектування інтерфейсу була зібрана, проаналізована та класифікована інформація щодо об'єкта, що проектується. Проведено системний аналіз системи «користувач – розумний холодильник» у ході якого було розроблено компонентно-функціональну та функціонально-об'єктну структури.

У новому інтерфейсі були реалізовані чотири основні функції, які були б корисні у використанні розумного холодильника:

- функція сканування штрих-коду та дати виробництва товару, дані про який поміщаються в базу даних холодильника (база даних холодильника передбачає вже внесену фірмою розробником інформацію про продукти, а саме інформацію про кількість днів споживання продукту при різних температурах);

- функція сканування еко-системи холодильника, яка за допомогою вбудованих всередину холодильної камери датчиків сканує холодильник на наявність шкідливих бактерій, які могли утворитися через неправильне зберігання продуктів та попереджає про це користувача;

- функція зміни температури яка дозволяє змінювати температуру не тільки у холодильній та морозильній камерах, а й окремо на кожній полиці;

- функція еко-режиму, яка знижує споживання електроенергії на 30%.

Всі ці функції знаходяться у головному меню холодильника наведеному на рис. 1.



Рисунок 1 – Головне меню розумного холодильника

Інтерфейс був розроблений за допомогою мови C# та середовища розробки Visual Studio.

На основі аналогу Samsung Family Hub та структурного аналізу був розроблений ергономічний дизайн розумного холодильника.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Samsung Family Hub [Electronic resource]. – Access mode: <https://www.samsung.com/us/explore/family-hub-refrigerator/overview/>

ПРОБЛЕМНО-ОРІЄНТОВАНИЙ ІНТЕРФЕЙС ЗАСТОСУНКУ “БЕЗПІЛОТНЕ ТАКСІ”

Безпілотний автомобіль – транспортний засіб, що здатен отримувати дані про своє середовище і здійснювати навігацію без участі людини, яка б ним керувала. Дві найсерйозніших перешкоди є на шляху розробників «безпілотників» – складність і дорожнеча технологій. Незважаючи на це, над розробкою безпілотних машин постійно працюють багато ключових гравців авторинку: General Motors, Tesla, Waymo, Uber, Lyft, Baidu [1]. А компанія Tesla планує запустити сервіс безпілотних таксі (БТ) вже у 2020 році.

І хоча на даний момент поява БТ на дорогах України майже неможлива через те, що відсутні умови які потрібні для роботи цих автомобілів, у майбутньому такий вид таксі здобуде популярність. Таксі подібного типу дуже перспективні, адже багато хто вважає безпілотний транспорт набагато безпечнішим. Також, компанії-розробники стверджують, що таке таксі буде коштувати набагато дешевше.

Так як існує багато прикладів застосунків для замовлення звичайного таксі, було розроблено інтерфейс застосунку для БТ, з попередньо проведеним аналізом аналогів. Основними вимогами до створення інтерфейсу були:

- можливість обрати початковий та кінцевий пункт призначення, дату та час поїздки;
- можливість робити декілька замовлень, редагувати та скасовувати їх;
- можливість підключення банківської картки для оплати замовлення;
- можливість керувати поїздкою (клімат-контроль, налаштування плеєра, додавання/видалення зупинок, зміна кінцевої адреси).

Розроблений підхід заснований на функціонально-структурній теорії ергатичних систем А.І. Губінського [2], узагальнюючої стратифікований підхід до подання систем. Для формалізації знань про систему “пасажи́р-машина-середовище” було проведено її компонентний та морфологічний аналіз. В процесі компонентного аналізу було розроблено компонентно-функціональну структуру, що дозволило виявити: користувачів ЛМІ (пасажир); контекст використання інтерфейсу; проблемні ситуації, які можуть виникати у процесі діяльності користувача. В процесі морфологічного аналізу було розроблено функціонально-об’єктну структуру,

що дозволило визначити сценарії дій користувачів у робочих, чи проблемних ситуаціях, та виявити номенклатуру елементів ЛМІ, що мають відображати необхідну інформацію.

Таким чином, проведений системний аналіз, та аналіз аналогів визначили основні функціональні вимоги до універсального БТ, та ЛМІ, що проектувався (рис.1).



Рисунок 1 – Фрагменти інтерфейсу програми

Практична цінність розглянутого підходу та розробленого на його основі ЛМІ, полягає у досягненні ефективності та комфортності під час керування БТ, мінімізації зусиль та часу для замовлення таксі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. УкрПрокат/блог Безпілотні авто. Перспективи автомобіля без водія [Електронний ресурс]. – Режим доступу: <https://ukr-prokat.com/blog/bezpilotni-avto-perspektivu-avtomobilya-bez-vodiya.html>
2. Информационно-управляющие человеко-машинные системы: исследование, проектирование, испытание. Справочник / А.Н. Адаменко, А.Т. Ашеро, И.Л. Бердников и др.; под общ. ред. А.И. Губинского. – М.: Машиностроение, 1993. – 528 с.

ЛЮДИНО-ОРІЄНТОВАНИЙ ПІДХІД ДО ПРОЕКТУВАННЯ ІНТЕРФЕЙСУ СИСТЕМИ КЕРУВАННЯ СВІТЛОФОРАМИ

Інтенсивне зростання кількості автомобільного транспорту, особливо в містах, постійно підвищує актуальність заходів, спрямованих на зниження негативних явищ, викликаних автомобілізацією. Збільшення завантаження вулично-дорожньої мережі знижує ефективність використання транспорту.

Для зниження негативного впливу автомобілізації використовується система керування світлофорами (СКС).

Так як відсутня єдина, загальновизнана технологія проектування людино-машинних інтерфейсів (ЛМІ), ціллю даного дослідження є розробка підходу до проектування міської системи керування світлофорами.

Аналогів СКС дуже мало і інформації про них теж. Умовно, за рівнями керування, диспетчерів СКС можна розділити на два типи: тих, що відповідають за управління системою світлофорів на закріплених за ними ділянках, та головного диспетчера, який має доступ до інформації про стан на ділянках всього міста, може відстежувати її, та у разі необхідності корегувати керування на усіх ділянках.

Предметом дослідження у даній статті є інтерфейс головного диспетчера СКС.

Розроблений підхід заснований на функціонально-структурній теорії ергатичних систем А.І.Губінського [1], узагальнюючої стратифікований підхід до подання систем. Для формалізації знань про систему “диспетчер-системне-середовище” було проведено її компонентний та морфологічний аналіз. В процесі компонентного аналізу розроблена множина компонентно-системних, компонентно-елементних, компонентно-функціональних та компонентно-режимних структур, що дозволило виявити: типи диспетчерів ЛМІ (диспетчер або головний керівник); контекст використання інтерфейсу; проблемні ситуації, які можуть виникати у процесі дорожнього руху.

В процесі морфологічного аналізу розроблена множина функціонально-часових, функціонально-об’єктних та функціонально-технологічних структур, що дозволило визначити сценарії дій диспетчерів у проблемних ситуаціях, та виявити номенклатуру елементів ЛМІ, що мають відображати необхідну інформацію.

Таким чином, проведений системний аналіз, визначив основні

функціональні вимоги до СКС, та ЛМІ, що проектувався (рис.1).

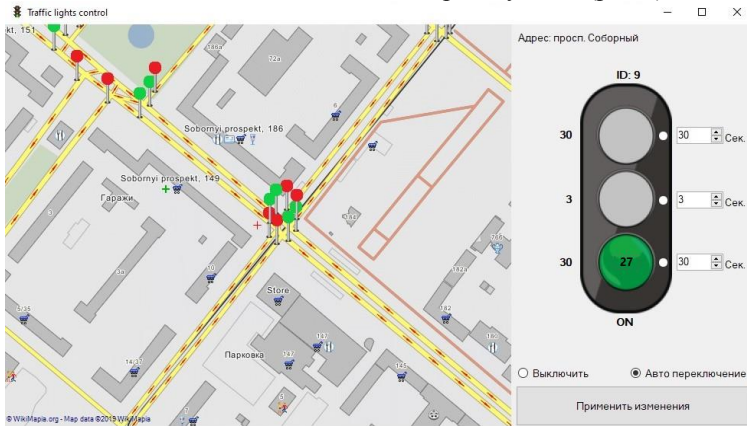


Рисунок 1 – Фрагмент інтерфейсу програми

СКС надає диспетчеру наступні можливості:

- вибір необхідного світлофора на інтерактивній мапі;
- вмикання і вимикання світлофора;
- фіксування певного кольору;
- зміна тривалості світіння будь-якого з кольорів.

Дана програма була створена в середовищі розробки Visual Studio 2019 мовою програмування C#.

Практична цінність розглянутого підходу та розробленого на його основі ЛМІ, полягає у підвищенні високорівневої ситуаційної обізнаності диспетчерів, ефективності їх діяльності, мінімізації зусиль і ризику помилок, зменшення психічного стресу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Информационно-управляющие человеко-машинные системы: исследование, проектирование, испытание. Справочник / А.Н. Адаменко, А.Т. Ашеро́в, И.Л. Бердни́ков и др.; под общ. ред. А.И. Губинского. – М.: Машиностроение, 1993. – 528 с.

ПРОГРАМНА РЕАЛІЗАЦІЯ АДМІНІСТРАТИВНОЇ ЧАСТИНИ ВЕБСАЙТУ «ІНФОРМАЦІЙНИЙ БЛОГ УНІВЕРСИТЕТУ»

Через те що увесь облік інформації переходить до цифрового формату, його адміністрування, а також адміністрування обмежень можливостей людей, які приймають участь у його заповненні, також має бути перенесене.

Це надасть зручний спосіб взаємодіяти з великими структурованими об'єктами інформації, зменшить необхідний час на проведення цих дій.

На сьогоднішній день усе йде на розподіл архітектури сайтів на front-end та backend для розділення відповідальності між рівнем представлення та рівнем доступу до даних.

Відповідно до цього, під час розробки вебсайту «Інформаційний блог університету» виникла задача щодо розробки адміністративної частини даного вебсайту, тобто засобу для автоматизації роботи по наповненню інформацією блогу для інформування студентів та робітників університету.

Для вирішення цього завдання було прийняте рішення створення інструменту для реалізації поточної структури взаємодії працівників університету для складання розкладу, оприлюднення новин університету та відомостей з оцінками студентів з предметів.

Надійність стоїть на першому місці під час розробки такого виду проєктів, а саме валідація введених даних для уникнення подальших проблем в роботі з ними, надання доступу до перегляду та взаємодії з інформацією після аутентифікації користувача та підтвердження прав для цих дій, шифрування паролів від облікових записів користувачів, тощо.

Адміністративна частина сайту була написана за допомогою фреймворку YII 2 на мові PHP, пакетного менеджера Composer, а також з використанням мови гіпертекстової розмітки HTML5 та каскадних таблиць стилів CSS3 для візуального відображення інформації. Для написання адміністративної частини було обрано кросплатформове інтегроване середовище розробки для PHP – JetBrains PhpStorm з вбудованою консоллю для взаємодії з розподіленою системою керування версіями файлів GIT.

Інтерфейс представляє в простому вигляді меню для вибору необхідних дій користувача з мапою для повернення на вищі рівні сайту. Безпека роботи з даними представлена набором ролей (адміністратор, деканат, завідуючий кафедрою, вчитель) для обмеження доступу для взаємодії з інформацією. Адміністратор також має право додавати нові ролі, нові обмеження та

змінювати існуючі.

На даний момент ієрархія залежностей та обмежень ролей має наступний вигляд:

Вчитель має права на:

- перегляд поточного розкладу його предметів;
- додавання новини за його предметами;
- створювання відомостей з оцінками з його предметів;

Завідуючий кафедрою має права на:

- додавання до системи вчителів своєї кафедри;
- перегляд розкладу усіх вчителів кафедри;

Деканат має права на:

- додавання до системи завідуючого кафедрою;
- додавання груп та студентів у них;
- додавання предметів та розроблення розкладу для кафедри.

Основна мова сайту та введення інформації є українська, але можлива зміна мови відображення візуальної частини сайту (російська, англійська), сама інформація залишається на написаній мові.

Середовищем функціонування програмного продукту обрана операційна система Windows 10.

Доступ до програмного продукту відбуватиметься через мережу Internet, що дає можливість користувачам, що знаходяться в іншій точці планети, переглянути необхідну інформацію.

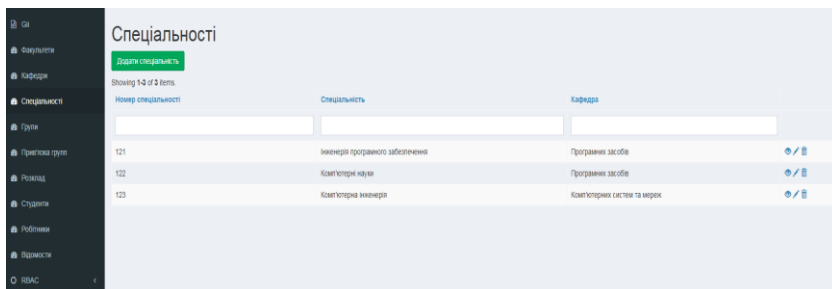


Рисунок 1 – Сторінка перегляду існуючих спеціальностей

Висновки

В роботі було розглянуто актуальність розробки адміністративної частини вебсайту «Інформаційний блог університету». Було запропоновано рішення щодо розробки цієї частини вебсайту. На даний момент програмний продукт знаходиться в процесі розробки та найближчим часом буде доступний для використання.

ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОВЕДЕННЯ ТУРНІРІВ З ФЕХТУВАННЯ

На сьогоднішній день розробка програмного забезпечення для проведення аматорських турнірів є достатньо важливою темою, адже маленькі товариства не можуть собі дозволити важкі та дорогі програмні засоби для комфортного проведення змагань. Велика кількість змагань проводять на дуже старих та не зручних програмах.

Таке програмне забезпечення допоможе більш швидше вести облік та більш комфортніше проведення змагань. Значними плюсами є швидкий обіг даних, зменшення помилок у розрахунках, що допоможе проводити турніри на більш високому рівні. Для проведення обліку на турнірі за допомогою програмного забезпечення потрібна усього одна людина.

Для організації даного програмного продукту необхідно розглянути основні вимоги до проведення турнірів. Основною метою розробки є реалізація програмного забезпечення для поліпшення проведення турнірів та їх організації, яке буде доступне усім зацікавленим.

Для досягнення цієї мети пропонується розробити базу даних учасників, тренерів та спортивних організацій.

Сутність «спортивна організація» має такі поля, як місто та назва організації (рис. 1). Якщо організація занесена до реєстру, подальше використання інформації про неї буде дуже комфортне та швидке (рис. 2).

Сутність «тренер» залежить від сутності «спортивна організація» та має поля: прізвища та назва організації, за якою закріплений тренер.

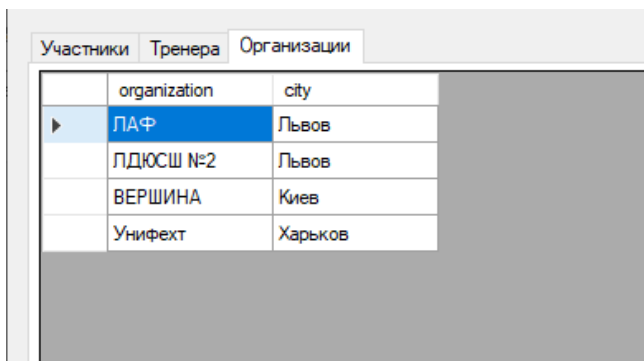
Сутність «учасник» збирає у собі усю потрібну інформацію про кожного спортсмена: ім'я, прізвище, стать, рік народження, розряд, організацію, за якою він закріплений, та список тренерів, які працюють з спортсменом.

Для усіх сутностей реалізовані такі функції як додавання, редагування та видалення учасників/тренерів/організацій.

Необхідно приділити увагу й інтерфейсу, який повинен бути зручним у використанні та інтуїтивно зрозумілим для користувача. Для здійснення повного проведення змагання був зроблений зручний інтерфейс, за допомогою якого дуже швидко можливо здійснити заповнення анкети учасника, перегляд реєстру зареєстрованих та швидке редагування їх.

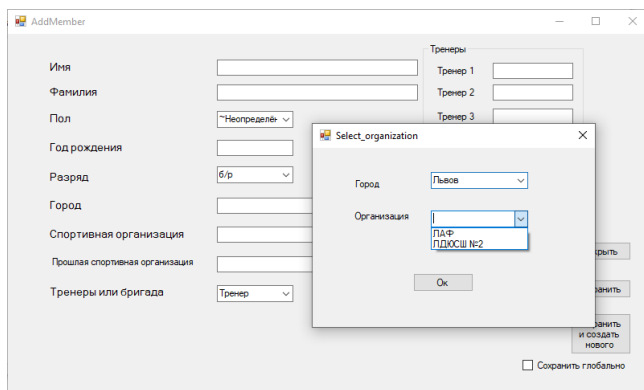
Програмне забезпечення вирішено розробити за допомогою мови

програмування C#, мови структурних запитів Structured query language. Розроблена програма є дуже зручною у використанні та зрозумілою кожному, хто користується типовими програмами обліку.



	organization	city
▶	ЛДФ	Львов
	ЛДЮСШ №2	Львов
	ВЕРШИНА	Киев
	Унифехт	Харьков

Рисунок 1 – Записи організацій у реєстрі



Имя:

Фамилия:

Пол:

Год рождения:

Разряд:

Город:

Спортивная организация:

Прошлая спортивная организация:

Тренеры или бригада:

Тренеры: Тренер 1 Тренер 2 Тренер 3

Сохранить глобально ☐

Select_organization dialog:

Город:

Организация:

ЛДФ

ЛДЮСШ №2

Ok

Рисунок 2 – Інтерфейс швидкого занесення організації до анкети

Висновки

В роботі було розглянуто основні вимоги для організації програмного засобу для проведення змагань за фехтування. На даний момент програмне забезпечення знаходиться в процесі розробки та найближчим часом може бути впроваджено в експлуатацію.

РОЗРОБКА КЛІЄНТ-СЕРВЕРНОГО ЗАСТОСУНКУ ДЛЯ ОБЛІКУ ЧАСУ СПІВРОБІТНИКІВ З ГНУЧКИМ ГРАФІКОМ РОБОТИ

Сучасний темп життя дуже насичений та на нього впливають багато зовнішніх обставин. Гнучкий графік роботи дозволяє людині не робити вибір між кар'єрою, хобі та сім'єю, а гармонійно поєднувати усі ці сфери. Тому, такий графік на ринку праці користується великим попитом серед працівників. Отже, і компанія з такими вимогами має можливість вибору найкращих кадрів серед усіх можливих. Особливо часто гнучкий графік роботи застосовується в ІТ-компаніях, рекламних агентствах, веб-студіях та в професіях, які пов'язані з творчою діяльністю. Дистанційна робота вигідна для молодих компаній, оскільки дозволяє значно заощадити кошти на утримання офісу.

Кількість часу витраченого на проєкт впливає на його собівартість, яку треба обґрунтувати замовнику. Тому дуже важливо знати чіткі цифри роботи кожного співробітника. Облік часу співробітника – основна проблема гнучкого графіку роботи або роботи дистанційно. Для вирішення цієї проблеми компанії використовують програмні застосунки. Але більшість готових рішень на ринку потребують щомісячної оплати за кожного працівника. Такі витрати є не вигідними для молодих компаній. Тому, у даній роботі розроблюється вебсайт для обліку часу співробітників, який повинен виконувати наступні функції:

- запис співробітником відпрацьованого часу;
- перегляд працівником умов своєї оплати праці;
- підрахунок своєї зарплатні;
- перегляд відпрацьованого часу всіх співробітників працівниками фінансового відділу;
- захист приватної інформації працівників;
- можливість повідомлення керівництва про роботу дистанційно;
- реєстрація нових користувачів;
- аутентифікація та авторизація в системі.

Для розроблення даного клієнт-серверного застосунку була обрана об'єктно-орієнтована мова програмування – Java. Застосунок може використовуватися під керування різних операційних систем, для яких існує реалізація JVM (JavaVirtualMachine).

Шаблон проектування MVC (ModelViewController) дозволить розробити

програмний продукт з незалежних модулів (рис. 1). Така архітектура дозволить легко додавати нову функціональність до застосунку в майбутньому або змінювати інтерфейс, без зайвих витрат часу. Сучасний фреймворкSpringBoot дозволить спростити та прискорити розробку даного продукту. Використання фреймворкуMaven дозволить без зайвих зусиль зібрати застосунок та підключити до проєкту необхідні бібліотеки.

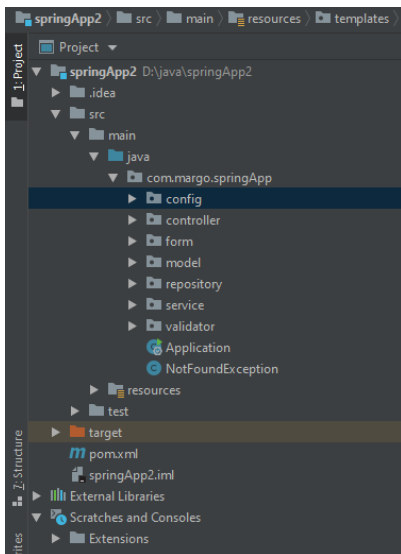


Рисунок 1 – Структура пакетів застосунку

Інтерфейс розроблюваного продукту досить простий та російськомовний. Такий підхід дозволить працівникам фіксувати відпрацьовані години вже з першого дня роботи. Розробка інтерфейсу виконується за допомогою мов HTMLта CSS, шаблонізатору Thymeleaf. Шаблонізатор дозволить клієнтській та серверній частині обмінюватися даними між собою.

Висновки

В роботі було розглянуто основні проблеми щодо обліку робочого часу співробітників. Було запропоновано рішення щодо вирішення основних проблем. На даний момент відповідний клієнт-серверний застосунок знаходиться в процесі розробки та найближчим часом буде доступний для використання.

ПРОГРАМНА РЕАЛІЗАЦІЯ КЛІЄНТСЬКОЇ ЧАСТИНИ ВЕБСАЙТУ «ІНФОРМАЦІЙНИЙ БЛОГ УНІВЕРСИТЕТУ»

Сучасний темп розвитку суспільства надзвичайно високий, це зумовлено потребою людини встигати за змінами, обробляти величезну кількість інформації. Тому інформаційні технології мають сьогодні пріоритетне значення в багатьох сферах діяльності й визначають майбутній розвиток суспільства. В інформатизованому світі, кожній школі, інституту, університету та іншим освітнім закладам зараз потрібно забезпечити, крім комп'ютеризації навчального процесу, реалізацію цільової програми інформатизації закладу освіти.

Завдяки комп'ютерним системам стало набагато зручніше вести інформацію в електронному вигляді. Наприклад: блог, розклад, відомість, консультації, інформація про викладача або інститут.

Перехід від «паперового» введення інформації до електронного, значно зручніший та надає швидкий і об'єктивний результат, дозволяє більш ефективно використовувати час.

Після порівняння існуючих рішень, було зроблено висновок, що актуальним завданням є розробка вебсайту «Інформаційний блог університету», а саме реалізація клієнтської частини вебсайту.

Метою роботи є створення користувацького інтерфейсу вебсайту «Інформаційний блог університету» для відображення інформації. Якісно спроектований інтерфейс користувача украй важливий для успішної роботи системи. Складний в застосуванні інтерфейс, як мінімум, приводить до помилок користувача. Іноді вони просто відмовляються працювати з програмною системою, не дивлячись на її функціональні можливості. Якщо інформація представляється плутано або непослідовно, користувачі можуть зрозуміти її неправильно, внаслідок чого їх подальші дії можуть привести до пошкодження даних або навіть до збою в роботі системи. Система відображає інформацію про: блоги університету та викладача, консультації та розклад в залежності від групи студента. Інтерфейс розроблювався для студентів та викладачів та повинен відповідати наступним вимогам: доступність; мінімалізм; чуйність; відповідність контексту; привабливість; ефективність.

Клієнтська частина реалізована з використанням фреймворку Yii 2, каскадних таблиць стилів CSS3 та мови розмітки HTML5, мова

програмування javascript, набір функцій jquery та bootstrap. Для реалізації вебсайту було обрано редактор Sublime Text 3, так як він є найпопулярнішою системою розробки веб-додатків.

Інтерфейс представлений в простому вигляді та в легкому користуванні для будь-якого користувача. Безпека додатку буде реалізована шляхом авторизації та аутентифікації користувачів, виділення різних ролей користувачів (викладач, студент, деканат).

Для створення сайту було обрано павутинну структуру. Таким чином користувач може легко перейти з будь-якої сторінки на будь-яку іншу.

Інформацію на вебсайті необхідно подавати українською мовою, а певні розділи – дублювати іншими мовами, згідно з особливостями регіону (російською, англійською). Для цього головна сторінка вебсайту має мати можливість переключення між цими мовами.

Середовищем функціонування програмного продукту обрана операційна система Windows 10.

Доступ до програмного продукту відбуватиметься через мережу Internet, що дає можливість користувачам, що знаходяться в іншій точці планети, переглянути необхідну інформацію. Вебсайт підтримує адаптивний дизайн (рис. 1), тому що практично кожна людина на планеті має смартфон. Зараз адаптивний дизайн – це вимога часу.

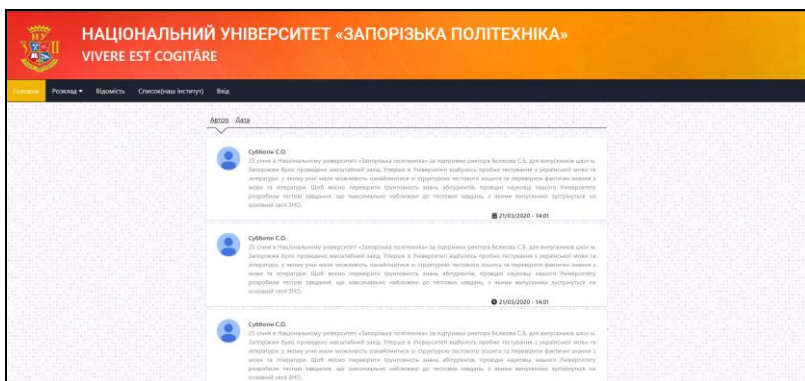


Рисунок 1 – Загальний вигляд головної сторінки сайту

Висновки

В роботі було розглянуто актуальність розробки вебсайту «Інформаційний блог університету». Було запропоновано рішення щодо розробки вебсайту. На даний момент програмний продукт знаходиться в процесі розробки та найближчим часом буде доступний для використання.

РОЗРОБКА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ З КОНТИНГЕНТОМ СТУДЕНТІВ

Для сучасних закладів вищої освіти (ЗВО), особливо державного підпорядкування, першочерговим завданням є ведення та моніторинг контингенту студентів. В будь-який період часу необхідно мати змогу отримати інформацію про кожного студента, форму навчання та вид фінансування, курс навчання, тощо. Для визначення цих даних необхідно контролювати усі накази, що стосуються процесу навчання даного суб'єкта. Також необхідно агрегувати дані для визначення групових показників за групами, спеціальностями, курсами та по ЗВО загалом. Для організації цього процесу в 2012 році державою було запроваджено інформаційно-комунікаційна технологія (ІКТ) – Єдина державна електронна база з питань освіти (ЄДЕБО), яка є автоматизованою системою збирання, реєстрації, оброблення, зберігання та захисту відомостей та даних з питань освіти [1]. ЗВО у цій системі є користувачем без права модернізації чи доопрацювання програмного забезпечення, тому, на жаль, не має можливості отримати статистичні звіти за різними типами запитів з ЄДЕБО, хоча по кожному студенту окремо інформація зберігається в історичному порядку.

Зважаючи на вищесказане, ЗВО вимушені розробляти власні ІКТ для автоматизації роботи з контингентом студентів, підтримки освітнього процесу, генерації звітів та отримання статистичної інформації.

Для розроблення ІКТ було вирішено:

- побудувати інформаційну модель руху студентського контингенту з прив'язкою до навчальних груп та структурних підрозділів університету;
- створити базу даних для збереження інформації, яка буде задовольняти вимогам моделі, та інтуїтивно зрозумілий інтерфейс для внесення, редагування та аналізу даних працівниками навчального відділу університету;
- створити інтерфейси та функціонал для інтеграції з ЄДЕБО та генерації стандартних звітів щодо руху контингенту студентів.

На першому етапі розробки потрібної системи було виділено основні сутності, а саме: студент, група, освітня програма (спеціалізація), спеціальність, кафедра, факультет, наказ, пункт наказу, зміст наказу. Базуючись на цих сутностях була розроблена концептуальна модель майбутньої системи.

Для вирішення завдання створення бази даних для збереження інформації про рух контингенту необхідно була обрана програма офісного пакету Microsoft Access 2019, що дозволило зберегти великий масив даних, а також зручно і швидко переглядати їх. Для обробки інформації щодо руху контингенту та зручного її перегляду було розроблено велика кількість форм та звітів, а також розроблено багато функціоналу за допомогою мови програмування VBA для заповнення стандартних звітів у вигляді Excel – таблиць. Для генерації звітів були підготовлені відповідні файли-шаблони, які включають у собі частину стандартної інформації кожного звіту (заголовки, макет) і програму встановлення стилів. В процесі генерації звітів виконується витягування потрібних даних і перенесення їх відповідно до вимог звіту. Пересування по файлу Excel по рядках виконується за допомогою ітератора, а по стовпцях за допомогою масиву із перебором імен потрібних стовпців.

Найбільшою проблемою у процесі ведення контингенту є дублювання інформації та необхідність фіксувати рух контингенту у декількох автоматизованих системах (ЄДЕБО та прикладних системах у різних підрозділах університету). З метою зменшення помилок, було прийнято рішення проводити щомісячну перевірку даних у розробленій системі з даними, внесеними у ЄДЕБО. Для цього необхідно отримувати усі дані щодо контингенту студентів з ЄДЕБО, що було реалізовано із використанням API ЄДЕБО. Отримані з ЄДЕБО дані завантажуються до відповідних таблиць ІКТ для автоматизації роботи з контингентом студентів для подальшої верифікації [2].

Висновки

В роботі було розглянуто актуальна проблема ефективної організації процесу моніторингу руху контингенту ЗВО. Було запропоновано рішення щодо автоматизації даного процесу. Отримана гнучка система дозволяє імпортувати дані з існуючих систем ведення контингенту, зручно опрацьовувати дані про рух здобувачів вищої освіти, контролювати процес відпрацювання наказів по студентам в ЄДЕБО, швидко отримувати усі типи статистичних звітів. Головним досягненням реалізованої ІКТ є те, що вдалося зменшити людський фактор у процесі обробки руху контингенту завдяки спостережанню і корегуванню дій користувача. Також з'явилася можливість надання доступу до даних робітникам ЗВО у межах університету завдяки розділенню даних та інтерфейсної частини системи.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. About edbo [Електронний ресурс]. – Режим доступу: <https://info.edbo.gov.ua/about/>
2. Брагина, Т. И. Информационная технология оценки достоверности

данных при интеграции web-ориентированных систем / Т.И. Брагина // Наукові праці Донецького національного технічного університету. Серія “Інформатика, кібернетика та обчислювальна техніка” (ІКОТ-2014). Випуск №1 (19) 2014. – Донецьк: ДВНЗ “ДонНТУ”. – 2014. – С.50-56.

УДК 004.42

Щетінін М.О.¹, Каплієнко Т.І.²

¹ студ. гр. КНТ-157сп НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

ПРОГРАМНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ВЕДЕННЯ ЕЛЕКТРОННИХ ВІДОМОСТЕЙ ДЛЯ ВЕБСАЙТУ «ІНФОРМАЦІЙНИЙ БЛОГ УНІВЕРСИТЕТУ»

Активний розвиток інформаційних технологій призвів до того, що більшість задач з обробки великих обсягів даних тепер виконуються виключно у електронному вигляді.

Більш того, наразі відбувається загальний перехід з використання локальних додатків, які працюють на комп'ютері користувача, до вебдодатків, які працюють на сервері та доступ до яких потребує використання веббраузера.

Це відбувається через те, що такі додатки, зазвичай, не потребують великої кількості ресурсів на комп'ютері користувача, є кросплатформними, тобто не потребують використання певного пристрою або певної операційної системи, та працюють у будь-якому веббраузері, один з яких встановлений на більшості сучасних електронних пристроїв за замовчуванням.

Додатковими перевагами використання вебдодатків є можливість отримати доступ до одних і тих самих даних з декількох пристроїв одночасно, можливість редагувати їх та переглядати зміни у реальному часі, майже повна впевненість у тому, що дані не можуть бути втрачені остаточно тощо.

Актуальність розробки програмного забезпечення для ведення електронних відомостей полягає у тому, що наразі у нашому університеті не існує вебсистеми ведення електронних відомостей, яка зробила би процес ведення та перегляду відомостей більш зручним як для викладачів, так і для студентів.

Метою роботи є вдосконалення процесу ведення відомостей. Система має функції для: введення даних у електронні відомості, відображення даних, введених у електронні відомості, друку відомостей.

Окрім цього, система повинна бути відкритою (доступ до неї будуть

мати як викладачі, так і у студенти) та повністю інтегрованою у вебсайт «Інформаційний блог університету».

Система створена з використання розповсюдженої мови сценаріїв загального призначення з відкритим вихідним кодом—PHP, високопродуктивного компонентного PHP фреймворка, призначеного для швидкої розробки сучасних вебдодатків—Yii, основою якого є модель MVC, та СКБД MariaDB, яка є відгалуженням реляційної СКБД MySQL, що розробляється спільнотою під ліцензією GPL та яка повністю сумісна з додатками, що використовують MySQL.

Додатково були використані системи Git (для контролю версій) та Composer (для швидкого додавання розширень та їх залежностей до фреймворку). В якості IDE було обрано PhpStorm, як одну із найпопулярніших систем для розробки вебдодатків.

Цей набір був обраний через те, що він складається з відкритих та дуже поширених технологій, які можна у майбутньому підтримувати та розгортати, використовуючи будь-яку платформу.

Інтерфейс представлений в простому та в легкому у використанні вигляді для будь-якого користувача (рис. 1).



Рисунок 1 – Загальний вигляд сторінки з оцінками всередині однієї з відомостей (альфа-версія)

Безпека додатку реалізована з використанням засобів аутентифікації користувачів та виділення різних ролей для них.

У системі використовується павутинна структура, тобто користувач може легко перейти з будь-якої сторінки на будь-яку іншу.

Середовищем функціонування програмного продукту може бути будь-яка система, яка має встановлений веббраузер.

Доступ до програмного продукту відбувається через мережу Internet, що дозволяє користувачам, які знаходяться в будь-якій точці планети, переглянути необхідну інформацію.

В якості хостинг-платформи для розгортання проекту було обрано сучасну хмарну хостинг-платформу – Google Cloud Platform через її гнучкість

у налаштуванні та надійність.

Висновки

В роботі було розглянуто актуальність розробки підсистеми ведення електронних відомостей для вебсайту «Інформаційний блог університету». Було запропоновано рішення щодо розробки підсистеми.

На даний момент програмний продукт знаходиться в процесі розробки та буде доступний для використання найближчим часом.

УДК 004.42

Каплієнко Т.І.¹, Галевська І.Д.²

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² студ. гр. КНТ-237сп НУ «Запорізька політехніка»

ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБСАЙТУ ДЛЯ ПРОСЛУХОВУВАННЯ МУЗИКИ З ВЗАЄМОДІЄЮ З КОРИСТУВАЧЕМ

Можливо, коли Інтернет-ресурси тільки розвивались, людина не замислювалась про створення вебсайту, але зараз вебсайти набирають шалену популярність. За допомогою такого ресурсу, як вебсайт, можна отримати швидко будь-яку інформацію, розвинути свій імідж, не виходячи з дому зробити купівлю, послухати музику, подивитися фільм, ознайомитися з роботами художників або фотографів, замовити додому їжу, подати резюме на роботу та заробляти.

Раніше не існувало вебсайтів, які б пропонували прослухати музику онлайн у будь-який час. Зараз існує безліч сайтів, наприклад, як «Google Play Music» [1] або «spotify.com» [2], але вони мають вагомні недоліки, як реклама та платний контент або недоступність у деяких регіонах країни, що не дуже зручно для користувача.

Тому було вирішено зробити вебсайт для прослуховування музики з взаємодією з користувачем.

Такий вебсайт може бути корисний як для музикантів, щоб висловити свій внутрішній світ та донести до слухачів своє мистецтво, так і для шанувальників будь-якого жанру, щоб слідкувати за новинами, як розвивається музика та що нового випускають виконавці.

Дуже важливо при створенні музичного сайту враховувати можливість постійного оновлення музичних композицій, особливо популярних.

Досить важливим етапом є дизайн, сайт повинен мати зручний, орієнтований для будь-якого користувача інтерфейс. Також сайт повинен мати категорії за різними жанрами, зберігати та виводити на екран інформацію про аудіофайли.

Клієнтська частина повинна включати в себе зміну шкали гучності, перемикавання та прослуховування бажаних треків.

Також користувач має можливість зареєструватись та увійти в уже існуючий акаунт, вводячи логін та пароль.

Одним з найнеобхідніших модулів є модуль пошуку (рис.1). Знайти потрібного музиканта або композиції користувач повинен швидко, пошук повинен виконуватися за назвою композиції, за виконавцем, роком видання, альбомом, тощо.

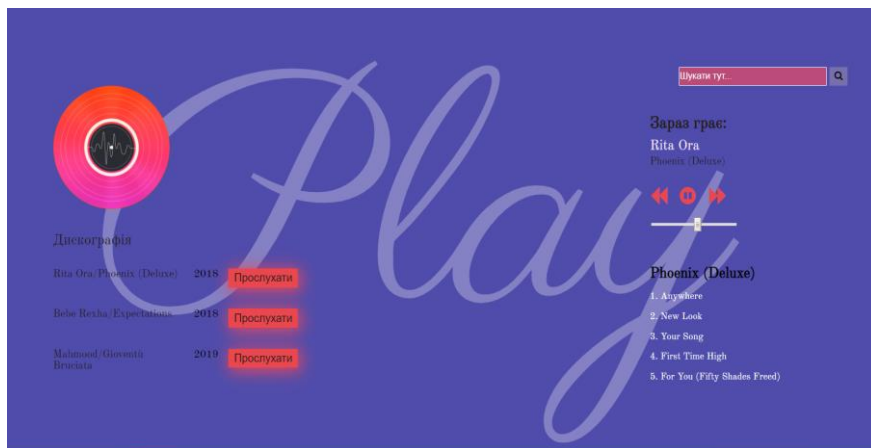


Рисунок 1 – Інтерфейс взаємодії з сайтом

Якщо говоримо про ідеальний музичний сайт, тоді на такому ресурсі повинна бути інформація про альбоми та їх рік випуску. Також має бути можливість прослуховувати цілий альбом виконавця або групи.

Адміністративна частина повинна передбачити можливість додавати, редагувати та видаляти дані про музику, додавати, редагувати та видаляти користувачів. Адміністратор повинен мати можливість виконувати авторизацію, здійснювати перегляд інформації користувачів, музики, категорій.

У якості інструментів для розроблення вебсайту для прослуховування музики було обрано об'єктно-орієнтовану скриптову мову програмування JavaScript, стандартну мову розмітки Hyper Text Markup Language, каскадні таблиці стилів Cascading Style Sheets, скриптову мову програмування Hypertext Preprocessor та систему управління контенту WordPress.

Висновки

В роботі було розглянуто актуальність розробки вебсайту для

прослуховування музики. Був опрацьований функціонал для поліпшення користування сайтом. На даний момент вебсайт знаходиться в процесі розробки та найближчим часом може бути впроваджено в експлуатацію.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Google Play Music [Електронний ресурс] – Режим доступу: <https://play.google.com/store/music>
2. Spotify [Електронний ресурс] –Режим доступу:<https://www.spotify.com>

УДК 004.415.25

Льовкін В.М.¹, Мироненко Н.В.²

¹канд. техн. наук, доц. НУ «Запорізька політехніка»

² студ. гр. КНТ-116 НУ «Запорізька політехніка»

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КЕРУВАННЯ РОБОТОЮ СТУДЕНТСЬКОГО ГУРТОЖИТКУ

Керування гуртожитком – багатоступеневий і всеохоплюючий процес. Він потребує роботи з великою кількістю даних. Зазвичай у гуртожитках усю інформацію зберігають на паперових носіях, що негативно впливає на процес її вивчення та обробки. Для передачі документів найчастіше використовується електронна пошта, однак це також не завжди зручно, бо повідомлення може загубитися у десятках інших листів.

Усі ці недоліки може вирішити наявність спеціального програмного забезпечення, яке могло б зберігати необхідні дані, обробляти запити користувача та видавати потрібну інформацію.

Розроблені до сьогоднішнього дня системи керування студентськими гуртожитками вирішують лише певні аспекти проблеми. Наприклад, призначені лише для формування звітів гуртожитку або займаються виключно етапом поселення. Але для повноцінного керування цього замало і потрібно створити програмне забезпечення, яке змогло б об'єднати усю необхідну функціональність у єдину систему.

Розглянувши проблему керування роботою гуртожитку, можна виділити наступні вимоги до розроблюваної системи:

- авторизація користувачів;
- реєстрація студентів завідувачем;
- поселення/перепоселення / виселення студентів;
- пошук студентів за різними параметрами;
- перегляд інформації про кімнати;
- облік руху товарно-матеріальних цінностей;

- оформлення звітів;
- реєстрація/ видалення працівників гуртожитку;
- редагування особистих даних завідувача, майстра, студента;
- отримання та ведення журналу заявок для майстра;
- відслідковування протермінованих замовлень майстром;
- виконання поселення / перепоселення студентом;
- виклик майстра студентом;
- отримання інформації про сплату за проживання в гуртожитку;
- зв'язок із завідувачем гуртожитку.

Для розробки програмного забезпечення керування роботою гуртожитку було обрано:

– для серверної частини – мову програмування PHP: PHP – мова програмування, яка найбільш поширена в сфері веб-розробки, на даний час PHP є одним з лідерів серед серверних мов програмування, що застосовуються для створення динамічних веб-сайтів і веб-додатків[1];

– для створення динамічних ефектів на сторінці –мову JavaScript;

– для формування структури веб-сторінки –мову розмітки HTML;

– для візуального оформлення сторінки – CSS;

– для створення якісної адаптивної верстки – інструмент Bootstrap;

– для підтримки комунікації між завідувачем, працівниками, студентами

– систему Callbell;

– у якості веб-сервера – Apache HTTP Server;

– у якості бази даних – MySQL;

– для адміністрування бази даних – інструмент phpMyAdmin;

– у якості бібліотеки для роботи з базою даних – PDO;

В основі розробленої системи лежить архітектурний шаблон MVC.

До компонентів MVC входять модель (відповідає за дані і визначає структуру програми), вид (визначає зовнішній вигляд програми і способи її використання), контролер (відповідає за зв'язок між моделлю і видом) [2].

Приклад інтерфейсу системи наведено на рис.1.

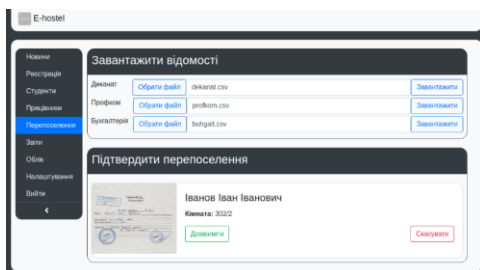


Рисунок 1 – Сторінка «Перепоселення» в особистому кабінеті завідувача

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Язык программирования PHP [Електронний ресурс]. – Режим доступу: <https://web-creator.ru/articles/php>.

2. TheMVCPatternandPHP, Part 1 [Електронний ресурс] –Режим доступу: <https://www.sitepoint.com/the-mvc-pattern-and-php-1/>.

УДК 004.921

Льовкін В.М.¹, Мяло О.Є.²

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² студ. гр. КНТ-116 НУ «Запорізька політехніка»

РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ГРАФІЧНОГО РЕДАКТОРА ДЛЯ ВІЗУАЛІЗАЦІЇ КОНЦЕПЦІЙ

Ефективність інтелектуальної роботи падає у відповідності з ростом кількості факторів, що з'являються у задачі.

Розроблене програмне забезпечення призначено для збільшення ефективності проведення інтелектуальної роботи за рахунок візуалізації та мінімізації втрат концентрації уваги.

Візуалізація концепції дозволяє окремо зберігати частину контексту задачі, що спрощує вирішення інтелектуальної задачі як такої, що має меншу кількість факторів для врахування. Для розширення даної концепції програмний засіб підтримує механізм просторів, що дозволяє розширити деталізацію вже намальованої концепції.

Інтерфейс програми побудовано таким чином, щоб на взаємодію з програмою витрачалося якомога менше концентрації уваги. Для цього створено спеціальний елемент «мультітул», для вибору інструментів малювання та взаємодії з намальованими об'єктами (рис. 1). Використання такого елемента дозволяє користувачу завчити розміщення кожної секції та потім обирати необхідний інструмент тільки за допомогою жестів.

Функції розробленого програмного забезпечення:

- «примітивне» малювання;
- малювання фігур;
- механізм просторів;
- збереження результату.

Для реалізації програми було використано такі інструменти:

- об'єктно-орієнтована мова C#;
- платформа Xamarin;
- бібліотека SkiaSharp для псевдо-векторного малювання;

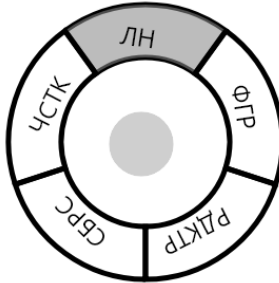


Рисунок 1 – Елемент для вибору інструментів

– бібліотека Newtonsoft.Json для роботи за даними у форматі JSON.

Xamarin – це платформа, що дозволяє виконувати C# і .Net код на різних апаратних платформах. Інтерфейс будується за допомогою Xamarin.Forms та XAML-подібної мови розмітки інтерфейсу. Її було обрано через наступні фактори. У випадку нестачі інструментів, що надає Xamarin, можна використовувати платформо-специфічні засоби, що значно зменшує недоліки, відносно нативної розробки під окрему платформу [1]. Застосування Xamarin дозволяє використовувати один і той самий C# код програмної логіки на різних платформах. Це створює перспективи з використання розробленого програмного засобу як на комп'ютерах, так і смартфонах та планшетах.

SkiaSharp – це C#-бібліотека, що базується на рушії з відкритим вихідним кодом Skia, що має апаратну оптимізацію на декількох платформах, таких як Windows, Android, iOS та інших, що робить його перспективним засобом для застосування разом з платформою Xamarin, логіка роботи графіки у Skia заснована на векторних принципах: малювання відбувається на полотні, щодо якого можна виконати операції трансформації – зміщення, масштабування, повороту та деякі інші за допомогою матриці трансформування, а отриманий рисунок можна зберегти у форматі SVG [2].

Програмне забезпечення було побудовано у відповідності з архітектурним шаблоном MVP (Model-View-Presenter), який є різновидом шаблона MVC. Різниця між цими шаблонами полягає у більшому відокремленні Моделі від Представлення, що дозволяє зменшити кількість логіки у Представленні [3].

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. What is Xamarin? [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin>
2. SkiaSharp Graphics in Xamarin Forms [Електронний ресурс]. – Режим

доступу: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/user-interface/graphics/skiasharp/>

3. Model-View-Controller в .Net [Електронний ресурс]. – Режим доступу: <http://rsdn.org/article/patterns/modelviewpresenter.xml>

УДК 004.415.25

Льовкін В.М.¹, Чорнобель Д.С.²

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² студ. гр. КНТ-116 НУ «Запорізька політехніка»

ПРОГРАМНА РЕАЛІЗАЦІЯ TELEGRAM-БОТА ДЛЯ ПІДТРИМКИ СТУДЕНТІВ ФАКУЛЬТЕТУ КОМП'ЮТЕРНИХ НАУК

В наш час багато шкіл, коледжів, університетів по всьому світу використовують мобільні, комп'ютерні або веб-застосунки, якими є як прості електронні довідники розкладу занять або розміщення аудиторій, які допомагають та підтримують студентів у навчанні, так і комплексні системи для обліку студентів та організації навчального процесу.

Зважаючи на популярність месенджеру Telegram серед студентів Національного університету «Запорізька Політехніка», а також його використання для ведення каналу кафедри програмних засобів, було вирішено розробити Telegram-бот, який надає студентам можливість знаходити аудиторію, переглядати розклад занять, отримувати матеріали за курсами за необхідності, реєструватися на заходи, отримувати дані про власні оцінки, а також контактну інформацію викладачів та відділів університету.

Викладачів програмний застосунок забезпечує функціями занесення оцінок або академічних заборгованостей студентам, та відправлення студентам методичних матеріалів або оповіщень.

Працівникам деканату програмний застосунок надає змогу встановлювати рейтинги старостам, відправляти їм повідомлення або файли, а також отримувати сформований рейтинг успішності студентів.

Студентського декана Telegram-бот забезпечує функціями керування даними старост груп студентів, ведення обліку боржників профкому, формування розкладу заходів та ведення обліку їх учасників, та виставлення додаткових балів студентам за проявлену активність.

Для розробки програмного забезпечення, яке реалізує вищезазначені функції, було використано такі засоби:

- мова програмування Python;
- pyTelegramBotApi – бібліотека, що реалізує Telegram Bot API;

- веб-фреймворк Flask;
- система керування базами даних PostgreSQL;
- бібліотека SQLAlchemy для роботи з системами керування базами даних з використанням технології ORM.

Для розгортання програмного застосунку на сервері було обрано PythonAnywhere, що є інтегрованим онлайн середовищем розробки та хостинговим сервісом для проєктів на Python та на основі його поширених веб-фреймворків [1].

Для реалізації здатності програмного застосунку приймати повідомлення від користувача було використано вебхуки. Структуру проєкту програми було організовано з використанням засобів, властивих мові програмування Python, а саме шляхом поділу проєкту на модулі (файли з розширенням .py, які містять у собі використовувані функції, класи, константи тощо) та пакети (спосіб структурування простору імен модулів; каталоги, які містять у собі модулі або інші каталоги, та включають в себе файл init.py) [2].

Інтерфейс Telegram-боту представлено на рис. 1.

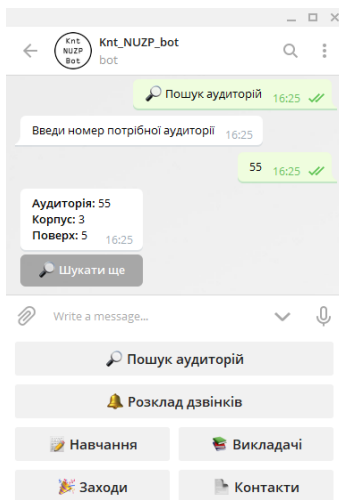


Рисунок 1 – Інтерфейс Telegram-боту

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. PythonAnywhere [Електронний ресурс]. – Режим доступу: <https://djangofriendly.com/hosts/pythonanywhere/>
2. Modules [Електронний ресурс]. – Режим доступу: <https://docs.python.org/3/tutorial/modules.html>

РОЗРОБКА МОБІЛЬНОЇ 3D-ГРИ ЗА ДОПОМОГОЮ СЕРЕДОВИЩА UNITY

У сучасному світі існує дуже велика індустрія розваг, яка в свою чергу складається з індустрії фільмів, серіалів, коміксів і звісно ігор. Ігри займають увагу великої кількості людей, тому є одним з найзручніших засобів для первинного входження в галузь розроблення програмного забезпечення. На даний момент існують потужні програмні засоби, які дозволяють виконувати процес розробки досить зручно, при цьому отримуючи в результаті достатньо якісні рішення. Окрім того ігри можуть використовуватися як зручний засіб для подальшої монетизації ідеї, а також для сумісного розповсюдження інших ідей розробника за рахунок створення комп'ютерної гри, що привертає увагу.

Ідея гри, яка надає користувачу можливість в ігровій формі вирішувати логічні задачі, засновані на проходженні різноманітних рівнів, була реалізована у програмному забезпеченні «WorldDestroyer», що представляє собою гру, яка складається з наступних структурних елементів і відповідних функціональних можливостей:

- різноманіття анімованих меню, що надають змогу навігації по можливостям гри, таких як обирання потрібного рівня, включення/виключення ігрової музики, та взагалі усіх звуків в грі, запуск безкінечного режиму, відображення зароблених очок тощо;

- система рівнів, де користувач не може починати наступний рівень перед тим, як не завершив попередній;

- система керування гравцем за допомогою простого натискання на екран і його плавного переміщення у задану точку;

- систему урахування поразок та перемог гравця;

- різноманітні перешкоди на шляху гравця, такі як повільні та швидкі платформи, турелі з різним ступенем наведення на гравця та ракети, що намагаються наздогнати гравця.

Реалізація зазначених функцій передбачала створення класів, які інкапсулюють у собі логіку програми, а також взаємодію з базою даних середовища. У результаті для розроблення програмного забезпечення було використано такий інструментарій:

- об'єктно-орієнтована мова програмування C#;

- кросплатформне середовище розробки комп'ютерних ігор Unity [1],

якефактично дозволяє виконувати одночасну розробку для усіхіснуючих мобільних пристроїв на всіх популярних платформах;

– Blender – це пакет для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, вимальовування, пост-обробки відео, а також створення відеоігор [2];

– SubstancePainter– програма для створення та накладання текстур на тривимірний об'єкт.

Зокрема, використані засоби розробки та прийоми продемонстровано на фрагменті інтерфейсу, наведеного на рис.1. Даний фрагмент демонструє створені тривимірні моделі з накладеними на них текстурами, інтерфейс користувача, що дозволяє користуватися функціями гри, та набір елементів, що складають один рівень гри.



Рисунок 1 – Представлення ігрової сценимобільної 3D-гри за допомогою середовища Unity

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Хокинг, Дж.Unity в действии. Мультиплатформенная разработка на C# [Текст] / Дж. Хокинг. –СПб. : Питер, 2019. – 352 с.
2. Blender – About [Електронний ресурс]. – Режим доступу: <https://www.blender.org/about/>

VISUALIZATION OF ALGORITHMS USING REMOTE LABORATORIES

Remote laboratories are known, for example, The Grid of Online Laboratory Devices Ilmenau (GOLDi) (Germany), visualise control algorithms using a WEB camera aimed at the physical model of the control object, as well as using visual model animation. If at the input of the animation model we generate a stream of events specific to the object of study (user calls for the elevator, incoming loads for the warehouse), then the complexity of the algorithms of the simulated system increases due to the processing of these flows. Therefore, visualization of their implementation is useful.

In this paper, we propose an animation model for the Warehouse physical model of the GOLDi laboratory, synchronized with the physical model. The model was written using SVG and JavaScript, supplemented by a generator of stochastic cargo arrangement in Warehouse. The model is animated on the basis of Warehouse operations (take / move / put cargo). The software implements an algorithm for sorting cargo into the Warehouse according to the principle of playing “15” with random initial placement and preliminary assessment of the achievement of sorting goals. A greedy graph search algorithm was used to solve the problem. Screens animations of the initial and final placement of cargo in the Warehouse are shown in the figure, and the sorting process video is available at <https://www.youtube.com/watch?v=yKwltVF6fMI>

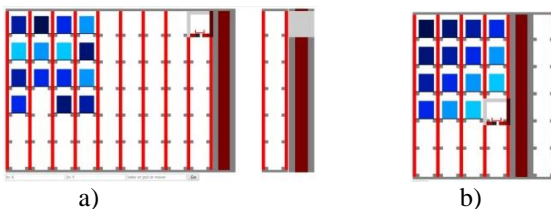


Figure 1 – Screens of the initial (a) and final (b) placement of cargo in the Warehouse

The operation of the algorithm was tested by the GOLDi laboratory at the Technical University of Ilmenau.

УДК004.9

Дейнега Л.Ю.¹, Басанець М.І.², Діденко А.Є.³, Камєнський Д.В.³

¹ старш. викл. НУ «Запорізька політехніка»,

² студ. гр. КНТ-117 НУ «Запорізька політехніка»,

³ студ. гр. КНТ-127 НУ «Запорізька політехніка»,

СКЛАДАННЯ РОЗКЛАДУ РОБІТ ДЛЯ ЗАДАЧ З ФУНКЦІЯМИ ШТРАФІВ

Проблема складання графіку на виконання деяких задач є вкрай актуальною через наявність пріоритетів у задач та штрафів за невиконання завдання. Іноді у групі людей декілька задач мають послідовно проходити через кожну людину, і тоді цікавить оптимальний час виконання цих задач. В загальному вигляді ці задачі описуються верстатами та часом опрацювання однієї деталі (задачі) на ньому і відомі як задачі теорії розкладів, або «Schedulingjobs».

Розглянутий в даній роботі алгоритм описує типову задачу з теорії розкладів – складання розкладу робіт для задач з функціями штрафів. Ця задача відома як задача Джонсона з одним верстатом, названа на честь вченого, що перший описав оптимальний алгоритм її вирішення. Алгоритми, що були запропоновані для вирішення даної задачі до Джонсона, працювали не менш як з квадратичною асимптотикою ($O(n^2)$). Алгоритм, розглянутий в даній роботі, працює з асимптотикою ($O(n \log n)$).

Задача Джонсона з одним верстатом – задача створення оптимального розкладу опрацювання n деталей на єдиному верстаті, якщо i -та деталь опрацьовується на ньому за час t_i , а за t секунд очікування до опрацювання цієї деталі платиться штраф $f_i(t)$.

Таким чином, задача полягає в тому, щоб знайти такий порядок опрацювання деталей, що розмір штрафу мінімальний. Якщо через p_i позначати номери деталей, то отримаємо наступну суму штрафу:

$$F(n) = f_{n_1}(0) + f_{n_2}(t_{n_1}) + \dots + f_{n_m} \left(\sum_{i=1}^{n-1} t_{n_i} \right) \quad (1)$$

Вирішимо цю задачу в випадку, якщо усі $f_i(t)$ лінійні. Після цього прийому вирішення задачі можна використовувати для інших видів функцій. Нехай функція штрафу має вигляд:

$$f_i(t) = c_i \cdot t, \quad (2)$$

де c_i – невід’ємні числа. В цих лінійних функціях вільний член дорівнює

нулю.

Зафіксуємо деякий розклад – перестановку P . Зафіксуємо деякий номер $i = 1 \dots n - 1$, і нехай перестановка P' дорівнює перестановці P , в якій замінили місцями i -ий та $i+1$ -ий елементи. Розглянемо, як саме змінився штраф F . Легко побачити з формули (1), що зміни відбулися лише в доданках з номерами i та $i+1$.

Якщо розклад P є оптимальним, то будь-яка зміна призводить до збільшення штрафу (або збереження значення штрафу). Таким чином, оптимальний розклад можна отримати, просто відсортувавши усі деталі за відношенням c_i до t_i в зворотному порядку.

Потрібно зазначити, що цей алгоритм було отримано так званим перестановочним прийомом: переставлено місцями два сусідні елементи розкладу, вираховано, наскільки при цьому змінився штраф і звідси отримано алгоритм пошуку оптимального розкладу.

Дана задача вирішується лише для трьох видів функцій штрафу: лінійної, експоненціальної та однакових монотонних функцій. Для всіх інших випадків перестановочний прийом не має сенсу та не працює (Теорема Лившиця-Кладова). В усіх трьох випадках шуканий оптимальний розклад може бути знайдений звичайним сортуванням, що використовує порівняння, тобто за $O(n \log n)$.

Даний алгоритм був реалізований мовою програмування C++, його робота протестована на декількох наборах даних. За результатами роботи алгоритму можна зробити наступні висновки.

Завдання складання розкладу робіт для задач з функціями штрафів можуть бути вирішене простим сортуванням послідовності за знайденим компаратором. Компаратор знаходиться за допомогою прийому перестановки двох сусідніх елементів послідовності та визначення внесених таким чином змін в шукані величини. Асимптотика алгоритму залежить від використаного методу сортування. Оптимальні методи сортування працюють за $O(n \log n)$, тому асимптотика розглянутих алгоритмів також $O(n \log n)$.

ПОРІВНЯННЯ АРХІТЕКТУР SPA ТА MPA В РЕАЛІЗАЦІЇ ВЕБ ЗАСТОСУНКІВ

Зі зростання популярності веб технологій і їхнього поширення у найрізноманітніших прикладних областях, архітектура розроблюваних веб застосунків стає все більш складною. Веб пройшов шлях від набору простих статичних сторінок, злінкованих між собою гіперпосиланнями, до складних хмарних застосунків, функціональні можливості яких можуть змагатися з десктопними програмами, а інколи і перевершувати їх. Перевагами таких веб застосунків стає їхня мобільність, доступ з будь якого місця та будь якого пристрою, зберігання даних та виконання обчислень у хмарі.

Для забезпечення сталої роботи складних веб застосунків необхідно використовувати архітектурний підхід, що надасть найліпшу швидкодію та продуктивність для кінцевого користувача. На сьогодні існує дві архітектури веб застосунків: односторінкові застосунки (Single page application або SPA) та багатосторінкові застосунки (Multi Page Application або MPA). Кожен з цих варіантів має свої особливості застосування, на які треба зважати на фазі проектування програмного забезпечення.

Розглянемо різницю між цими архітектурами та випадки, в яких кожна з архітектур дає свої переваги.

SPA дозволяє будувати застосунки, які за своєю поведінкою схожі на десктопні програми. Такі застосунки не вимагають перезавантаження сторінки в ході роботи. SPA дуже швидко реагує на дії користувача і є настільки інтерактивним, що у користувача складається враження роботи з десктопною програмою, він не відчуває затримок, необхідних для завантаження нових сторінок. Типовий SPA застосунок - це веб сторінка, що підвантажує дані та оновлює свій вміст без перезавантаження за допомогою мови JavaScript. Для побудови SPA використовують сучасні JavaScript фреймворки, такі як AngularJS, ReactJS, тощо.

Перевагами SPA є:

- розділення на front-end та back-end розробку;
- висока швидкодія завдяки тому, що більшість ресурсів (HTML, CSS, скрипти) завантажуються лише один раз при ініціалізації застосунку, після чого завантажуються лише дані;
- швидка розробка завдяки сучасним фреймворкам, які спрощують створення архітектури проекту і надають велику кількість готових елементів

для роботи;

- можливість легкого створення мобільного застосунку з тим же кодом на стороні серверу;
- застосунки можуть кешувати дані на локальних носіях та працювати в режимі оффлайн.

До недоліків SPA можна віднести наступне:

- складність SEO-оптимізації;
- довготривале початкове завантаження застосунку;
- потребують ввімкненого JavaScript в браузері користувача;
- менш захищені адже за допомогою міжсайтового скриптингу (XSS), злоумисники можуть впроваджувати додаткові скрипти на стороні користувача.

Багатосторінкові застосунки мають більш класичну архітектуру, в якій в результаті запитів на сервер повністю оновлюється та рендериться весь вміст сторінки, навіть у тому випадку коли нових даних не багато. Таким чином витрачається час на відображення одних і тих самих елементів, що впливає на швидкість та продуктивність. Зазвичай така архітектура використовується при необхідності відображення великої кількості інформації.

Можна виділити наступні основні переваги МРА:

- спрощення SEO оптимізації, адже архітектура МРА дозволяє легко оптимізувати кожен сторінку під пошукові системи;
- легка розробка, необхідний менший стек технологій, тобто розробка такого застосунку може бути дешевшою;
- більш звичний інтерфейс для кінцевого користувача.

В той же час у архітектури є і недоліки:

- довший час і складність розробки;
- зв'язані між собою front-end та back-end, складно розвивати їх паралельно;
- додаткова розробка для отримання можливості завантаження даних без перевантаження сторінки з використанням технології аїах;
- багаторівневий дизайн інтерфейсу користувача.

Отже розглянувши особливості кожної архітектури, можна зробити висновок, що кожна з них підходить для певного типу проєктів. Так SPA краще підходить для проєктів, в яких пошукова оптимізація не є визначальною, наприклад, SaaS застосунки або соціальні мережі. Для створення ж e-commerce застосунків або корпоративних сайтів краще підійде архітектура МРА.

УДК 004.9

Олійник А.О.¹, Федорченко Є. М.², Злиденний А.О.³

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² старш. викл. НУ «Запорізька політехніка»

³ студ. гр. КНТ-216 НУ «Запорізька політехніка»

ПРОГРАМНА РЕАЛІЗАЦІЯ АРКАДНОГО ЗАСТОСУНКУ ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ

Сьогодні роль мобільного телефону в житті людини важко переоцінити. За допомогою цього компактного приладу користувачі в будь-який момент можуть швидко зв'язатися зі своїми родичами і друзями, колегами по роботі, щоб дізнатися цікаву для них інформацію. При цьому не треба напружувати пам'ять і згадувати телефон старого знайомого, з яким вже давно не спілкувались, записна книжка телефона завжди під рукою. Багато крім контактів зберігають на телефоні ще масу іншої інформації - записують свої ідеї та думки, номери кредитних карт, пам'ятні дати, різні файли і т.д. Тобто телефон замінює відразу записну книжку і flash-накопичувач. А всілякі корисні програми, функції, опції якими оснащені сучасні мобільні телефони роблять це невелике за розміром пристрій настільки багатофункціональним.

Створення ігор – це трудомісткий процес, який вимагає багато часу і обдуманості роботи. Щоб створити хорошу гру на Андроїд або IOS потрібно докласти чимало зусиль, таке можна сказати і про створення програми. Все залежить від ідеї і обраного методу реалізації задуманого. На даний момент, при створенні ігор, майже всі розробники задіють ігровий движок і це істотно полегшує завдання

Даний 2D мобільний додаток призначена для розважального контенту, де користувач може з легкістю та з користю проводити свій вільний час. Робота полягає в реалізації 2D мобільного додатку для Android та IOS користувачів.

Програма розрахована на будь-якого користувача, який має мобільний телефон на базі Android та IOS. Гра має зручний та практичний інтерфейс, який має свої особливості, задній фон, який кожен раз буде змінюватись. Все це зроблено для надання гри гіпер-казуального характеру.

Розроблений мобільний додаток виконує всі функції, як для 2D гри аркадного характеру, а також інформацію користувачу про розробника. Програма має актуальні за функціональними можливостями характеристики та будову (робота на всіх мобільних пристроях Android та IOS платформ); підтримка роботи в ігровому движку Unity 2D/3D. Повна реалізація гри на Windows 10 (64-bit) Програма може бути перенесена на будь-який носій інформації (це дозволяється завдяки малому розміру її керуючого кода) і

завантажена з нього.

Мобільний застосунок оснащений всіма функціями, які мають 2D гіпер-казуальні ігри, тобто гра не має ніяких програмних помилок в робочому коді. Тестування мобільної гри “Великий Стрибок” було на мобільному телефоні Android платформи, гра працювала ідеально.

Зручність та легкість керування основним кубом роблять розроблену програму, доволі невибагливою до користувача, а наявність різних Assets роблять гру незвичайною та не схожую на своїх попередників в 2D аркадних іграх.

За програмними якостями та здібностями, щодо розробленого 2D додатку, можливо навести декілька схожих аналогів – “Flappy Bird” (2018 рік) та “Dunk Hit” (2019 рік), але вони мають компанії - виробників та мають великий розмір в пам’яті телефона.

УДК 004.9

Олійник А.О.¹, Федорченко Є. М.², Харченко А.С.³

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² старш. викл. НУ «Запорізька політехніка»

³ студ. гр. КНТ-116 НУ «Запорізька політехніка»

РОЗРОБКА МОДИФІКОВАНОГО ГЕНЕТИЧНОГО МЕТОДУ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ РОЗМІЩЕННЯ ОБ’ЄКТІВ ЕЛЕКТРОПОСТАЧАННЯ

Задача перспективного розвитку міської розподільної мережі електропостачання має на увазі визначення оптимального варіанту розвитку мережі, який враховує темпи розвитку міської інфраструктури. При цьому обирається варіант розвитку мережі, при якому за найменших затрат на її будівництво і експлуатацію буде забезпечений заданий рівень надійності та якості електроенергії, що передається. У зв’язку зі складністю та багатомірністю сучасних електропостачальних мереж, багатоваріантністю та багатокритеріальністю, задача обґрунтування перспективного розвитку міської розподільної мережі електропостачання є погано формалізованою, вимагає для свого розвитку дуже високих обчислювальних затрат, з практичної точки зору є важко розв’язуваною [1].

При проектуванні систем електропостачання різного призначення завжди мають місце обмеження генерального плану об’єкта проектування, технології виробництва і т.п. Виникає необхідність розробки нових математичних моделей і методів вирішення завдань, що враховують обмеження в нерівномірно розподілених електричних навантажень та в

довільній формі території, на якій проектується система електропостачання [2].

У зв'язку зі складністю поставленого комплексу завдань з обґрунтуванні раціональної конфігурації систем електропостачання, багато завдань не знайшли ще свого детального розгляду і вирішення. До цих завдань відносяться: завдання оптимального розміщення одиничного джерела живлення (ДЖ) з урахуванням довільних обмежень на місцевості в разі радіальної електричної мережі, завдання оптимального закріплення споживачів за декількома ДЖ. А також завдання оптимального розміщення декількох ДЖ (різних типорозмірів) і одночасного закріплення споживачів за цими джерелами живлення [3].

Для вирішення цієї проблеми було розроблено програмний продукт призначений для вибору раціональної конфігурації електромережі. Під конфігурацією системи електропостачання розуміють певне взаємне розташування елементів системи електропостачання, взаємозв'язок елементів в системі електропостачання, з'єднання елементів в форми певної структури [3]. Було виконано дослідження та програмна реалізація методів розміщення об'єктів електропостачання.

Розроблено програмний продукт для рішення задачі оптимального розміщення декількох ДЖ різних типорозмірів та закріплення за ними споживачів в системі електропостачання з використанням алгоритмів генетичного програмування. Запропоновано модифікацію простого генетичного алгоритму, на основі котрого створено програмний продукт. Розроблений ГА вирішує питання комбінаторної оптимізації у відношенні вибору оптимальної локації розміщення джерел живлення у розподіленій електричній мережі.

Під час тестування даного алгоритму було встановлено, що він дозволяє отримати найкращі рішення в період виконання перших 5-30 ітерацій, що є прийнятним показником. Проведена оцінка часу розрахунку в залежності від параметрів задачі. Показано, що для задач малої і середньої розмірності розроблений алгоритм забезпечує мінімальний час рахунку. Результати рішення задачі для конкретного прикладу демонструють перевагу генетичного підходу над методом повного перебору.

Отримані результати дозволяють запропонувати ефективні методи і алгоритми для вирішення широкого класу оптимізаційних задач, що виникають при обґрунтуванні раціональної конфігурації систем електропостачання. Використання результатів досліджень буде сприяти ефективній роботі розподільчих електричних мереж.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ravadanegh, S. On optimal multistage electric power distribution networks expansion planning / S. Ravadanegh, T. R. Roshanagh // International Journal of Electrical Power & Energy Systems. – 2014. – Vol. 54. – P. 487–497.
2. Hofmann, R. Siting and sizing of distributed energy storage to mitigate voltage impact by solar PV in distribution systems / R. Hofmann, S. Panuschka, A. Beck // Computers & Chemical Engineering. – 2019. – Vol. 128. – P. 246–260.
3. Syahputra, R. Distribution Network Optimization Based on Genetic Algorithm / R. Syahputra // Journal of Electrical Technology UMY. – 2017. – №1. – P. 20–33.

УДК 004.9

Степаненко О.О.¹, Федорченко Є. М.², Трошина Н.Є.³, Гончаренко Д.А.³,
Бородін О.О.⁴

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² старш. викл. НУ «Запорізька політехніка»

³ студ. гр. КНТ-216 НУ «Запорізька політехніка»

⁴ студ. гр. КНТ-726сп НУ «Запорізька політехніка»

ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ СТАНУ КОМП'ЮТЕРА

З розвитком сучасних локальних обчислювальних мереж (LAN) проблема управління і моніторингу мереж стає все більш нагальною. Все більше грошей доводиться витратити організаціям на підтримку нормального функціонування своїх локальних мереж. У наші дні складно уявити сучасну організацію або компанію, в якій би не було комп'ютерів, а комп'ютерна мережа є найбільш природним продовженням персонального комп'ютера. Вже перестали бути рідкістю і домашні мережі, які об'єднують мешканців одного під'їзду чи будинку. Звичайно, для управління мініатюрної домашньої комп'ютерною мережею на 5-10 комп'ютерів не потрібно спеціальних засобів, проте, вже сама поява такої мережі говорить про велике поширення комп'ютерних мереж і їх зростання.

Мережі ростуть, а, отже, разом з масою переваг несуть з собою і масу проблем, які пов'язані з управлінням. Основною проблемою, безсумнівно, є забезпечення працездатності мережі. Адже пересічному користувачеві не так важлива швидкість, як сталість і надійність роботи мережі.

Дана програмна система призначена для визначення характеристик ЕОМ, і передачі цієї інформацію на сервер для подальшої обробки. Це допомагає системному адміністратору комп'ютерної мережі оперативно

відстежувати порушення в роботі робочих станцій, при цьому він зможе це робити з сервера, що, безсумнівно, є перевагою. Робота полягає в реалізації на ЕОМ операцій моніторингу стану персонального комп'ютера.

Програма розрахована на недосвідченого користувача, не знайомої з програмуванням. Вона має зручний інтерфейс, який спрощує роботу з програмою, системою допомоги, призначеної для знайомства з роботою програми.

Розроблена програмна реалізація виконує діагностування ресурсів ПЕОМ (тестування CPU, E-IDE, RAM та CMOS), а також надає інформацію про багато інших ПУ (чипсет “материнської” плати, шину PCI/AGP, BIOS (ROM), відеосистему, FDD, “мишу”, клавіатуру). Програма має актуальні за функціональними можливостями характеристики та будову (робота на ПЕОМ з процесорами від I80386 SX до Pentium III / Athlon); підтримка роботи в ОС 2-х типів: MS DOS та MS WINDOWS XP. Програма може бути перенесена на будь-який носій інформації (це дозволяється завдяки малому розміру її керуючого кода) і завантажена з нього.

Програма оснащена, так званим, “захистом від дурня”, тобто вона не призводить до втрати або руйнування даних діючої ОС при будь-яких діях користувача, що тестує нею ПЕОМ.

Зручність та легкість керування оболонкою роблять розроблену програму, доволі невибагливою до користувача, а наявність вмонтованої сторінки допомоги значно підвищує її якість, як програмної розробки.

За програмними якостями та здібностями, щодо розробленої програми, можливо навести лише єдину програму-аналог – SpeedSYS.474 (2016 рік), але вона має розмір в 6 раз більший і тому за своїми якостями поступається розробленій.

УДК 004.9

Степаненко О.О.¹, Федорченко Є. М.², Трошина Н.Є.³, Гончаренко Д.А.³,
Брильов Є.С.⁴

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² старш. викл. НУ «Запорізька політехніка»

³ студ. гр. КНТ-216 НУ «Запорізька політехніка»

⁴ студ. гр. КНТ-215 НУ «Запорізька політехніка»

ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-САЙТУ РЕСТОРАНУ

Протягом останніх років інтернет поглинає всі можливі сфери діяльності. Сьогодні, починаючи будь-яку справу, необхідно одразу виділяти частину стартового бюджету на веб-сайт вашого проекту.

Сайт – найпростіший спосіб донести інформацію до сучасної людини. Ви здатні оперувати, як текстовим, так і медіа контентом для демонстрації свого продукту.

Сьогодні кожна людина, якій потрібно щось купити, або дізнатись будь-яку інформацію, перш за все, намагається знайти її в інтернеті.

Окрім реклами та зручності для споживача, сайт також має практичну цінність для власника бізнесу. Це перш за все – часткова, або повна автоматизація роботи вашого бізнесу. На прикладі цієї роботи, сайт дозволяє суттєво покращити наступні аспекти роботи:

- можливість онлайн бронювання столику;
- освітлення новин та акцій ресторану;
- ознайомлення гостей з актуальним меню закладу.

Розроблений сайт призначений для покращення контакту між закладом та відвідувачами шляхом спрощення отримання потрібної інформації останнім. Також продукт надає можливість власникам сайту збільшити швидкість роботи закладу та перекладає ряд функцій з персоналу на “комп’ютер”.

Сайт містить в собі наступні функції:

- можливість бронювання столика користувачем;
- відправка листа на електронну пошту користувачу, бронювання якого було підтверджено;
- можливість керування бронями з боку адміністратора;
- можливість керування списком страв закладу з боку адміністратора;
- для зручності реалізовано функцію пошуку по стравам та броням в панелі адміністратора;
- в якості запобіжного заходу, реалізовано шифрування пароля в базі даних.

В кінцевому варіанті, сайт дає можливість споживачу ознайомитись з новинами закладу, його меню, а також розкладом роботи. Також, споживач має можливість зробити бронювання, обравши стіл та зручний час.

Сайт відповідає всім вимогам сучасності, він коректно працює на всіх актуальних браузерях, на будь-яких девайсах з будь-якими розмірами екрану.

Також, окрім частини, призначеної для користувача, при розробці сайту, не забули про тих, хто буде сайт адмініструвати. Була розроблена проста та водночас зручна адміністраторська панель, де можна керувати існуючими бронями, а також додавати / видаляти / редагувати меню, розташоване на сайті. Адміністраторська панель захищена кроком авторизації. Логін та пароль адміністратора знаходиться в базі даних. Щоб додати нового користувача, або редагувати дані старого, необхідно мати доступ безпосередньо до бази даних. Реєстрацію нового користувача одразу на сайті була врахована недоречною з точки зору безпеки.

У кінцевому підсумку, було створено готовий до реалізації продукт, який в силу структурованого коду, можна швидко та без проблем підігнати під будь-який заклад.

УДК 004.93

Степаненко О.О.¹, Федорченко Є. М.², Трошина Н.Є.³, Гончаренко Д.А.³, Шлома Р.О.⁴

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² старш. викл. НУ «Запорізька політехніка»

³ студ. гр. КНТ-216 НУ «Запорізька політехніка»

⁴ студ. гр. КНТ-415 НУ «Запорізька політехніка»

РОЗРОБКА БАЗИ ДАНИХ ДЛЯ НАВЧАЛЬНОГО ВІДДІЛУ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЯ ГЕНЕРАЦІЇ ПЛАНУ НАВЧАННЯ

На даний момент навчальний відділ Національного університету «Запорізька політехніка» працює з достатньо великими масивами даних і більша частина рутинної роботи вже автоматизована і спрощена за допомогою сучасних персональних комп'ютерів і інших машин. Але ще не було створено такої програми, що дозволяла б працювати із графіками навчальних процесів у простому і зручному форматі, а саме: працювати з великою кількістю графіків в один момент, створювати їх, обробляти і на їх основі генерувати необхідні звіти.

Актуальність роботи полягає у тому, щоб створити програму, яка вирішує поставлені проблеми, які в повній мірі ще не були вирішені на даний момент.

Система забезпечує виконання таких функцій:

- додавання, редагування, видалення інформації про графіки;
- додавання, редагування, видалення інформації про навчальні роки;
- генерація графіків навчальних процесів на основі стандартної логіки їхнього створення;
- дублювання записів про графіки одного року для іншого;
- розбиття рядків запису «Графік» на зручні табличні, де їх можна зручно переглядати, відредагувати і зберегти зміни;
- підрахунок символів графіку навчального процесу і виведення їхньої кількості;
- просте і наочне прив'язування спеціальності до графіку і навпаки;
- генерація звіту «Навчальний план» для певного року вступу в одному з режимів генерації: для кожної спеціальності, для обраних спеціальностей і

для обраної однієї спеціальності на вибір для денної і заочної форми навчання. Генерація виконується для денної і заочної форми навчання окремо;

– генерація звіту «Графік навчального процесу» для певного року за одним з режимів генерації: повний АЗ (вносяться всі записи графіків за роком), для кожної спеціальності окремо (вносяться записи графіків за роком і спеціальністю), для обраних спеціальностей (вносяться записи графіків за роком і лише за обраними спеціальностями), для спеціальності (вносяться записи графіків за роком і лише для однієї обраної спеціальності). Генерація виконується окремо для денної і заочної форми навчання;

– автоматичне форматування файлу Excel звіту «Графік навчального процесу»: відмальовування сітки за вмістом (границь документу), відмалювання кривої початку весняного семестру, виставлення розміру комірок спеціальностей за вмістом (при повному режимі генерації).

Завдяки можливостям MS Access були створені зручні і прості форми для взаємодії із даними про графіки і паралельний контроль помилок користувача, які могли б виникнути через людський фактор.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Закон України «Про вищу освіту» : за станом на 1 липня 2014 р. : офіц. вид. / Верховна Рада України. – Київ : Парлам. вид-во, 2014. – 49 с. – (Закони України).

2. Методичні рекомендації щодо запровадження Європейської кредитно-трансферної системи та її ключових документів у вищих навчальних закладах [Електронний ресурс]. – Режим доступу: <https://kpi.ua/metod>.

3. Microsoft Access 2010 [Електронний ресурс]. – Режим доступу: <http://softcatalog.info/ru/programmy/microsoft-access-2010>.

4. Реляційна модель даних [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Реляційна_модель_даних.

5. Моделі даних. Ієрархічна модель даних. Мережна модель даних. Реляційна модель даних [Електронний ресурс]. – Режим доступу: https://geoknigi.com/book_view.php?id=589.

УДК 004.9

Качан О.І.¹, Федорченко Є. М.¹, Степаненко О.О.², Скачко Л.П.³,
Трошина Н.С.⁴, Гончаренко Д.А.⁴, Коломоець В.М.⁵

¹ старш. викл. НУ «Запорізька політехніка»

² канд. техн. наук, доц. НУ «Запорізька політехніка»

³ асист. НУ «Запорізька політехніка»

⁴ студ. гр. КНТ-216 НУ «Запорізька політехніка»

⁵ студ. гр. КНТ-136сп НУ «Запорізька політехніка»

ПРОГРАМНА РЕАЛІЗАЦІЯ КАРТОГРАФІЧНОГО МОДУЛЯ ДЛЯ ТРЕНАЖЕРА АРТИЛЕРІЇ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

Впродовж всього існування людства було актуальним воєнне ремесло. Важливою частиною війни є тактика ведення бойових дій. За неї, більшою частиною, відповідні офіцери. Ці військові займаються не лише організацією бойових дій, а й контролюють їх ведення. З розвитком різноманіття використання ресурсів й світової економіки у цілому, ймовірність й, відповідно, кількість воєнних дій зростає.

Саме через це, актуальність питання використання інструментів планування бойових дій навіть не повстає. Цифрове рішення здатне полегшити й додати різноманіття в цей процес, не кажучи вже про все більш широке використання технологій та можливість всебічної кастомізації на відстані догику. Враховуючи все вищесказане й загальний розвиток інформаційних технологій у цілому, необхідно створити модульну програмну систему з картографічним модулем який використовує технологію доповненої реальності. Найзручніше було б створити програмну систему у вигляді мобільного застосунку.

Мобільний застосунок має наступний функціонал:

- у якості початкових даних приймає введенні користувачем значення;
- налаштовує ландшафтне зображення використовуючи початкові дані;
- оператор-артилерист корегує масштаб ландшафту та координати розташування комплексних тривимірних елементів, які зображують структури будівель або військової техніки;
- у режимі реального часу розпізнає контрольну поверхню тренувального приміщення та оцінює її придатність для відображення у доповненій реальності тривимірних елементів;
- відображає дружній користувацький інтерфейс, простий для розуміння військовим офіцерам;
- у режимі реального часу виконує відображення 3D-структури ландшафту та оновлює її використовуючи початкові дані.

У найближчі часи планується розробка цілісної автоматизованої

системи яка буде мати тенденції впровадження до військової кафедри для навчання офіцерів. Після впровадження програмна розробка має окупитися за 2 роки, а також, завдяки використанню технології доповненої реальності має полегшити задачі військових офіцерів та артилеристів, при тому значно зменшуючи кількість помилок.

УДК 004.4'22

Миронова Н.А.¹, Ліпихін С.О.², Терлецький С.В.², Денисенко С.В.²

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² студ. гр. КНТ-138 НУ «Запорізька політехніка»

СУЧАСНІ ТЕХНОЛОГІЇ КОНТЕЙНЕРИЗАЦІЇ

Контейнеризація – це віртуалізація і ізоляція ресурсів на рівні операційної системи, яка дозволяє запускати застосунок і необхідний йому мінімум системних бібліотек в повністю стандартизованому контейнері, що з'єднують з хостом або чим-небудь зовнішнім по відношенню до нього за допомогою певних інтерфейсів.

Всі компоненти, необхідні для запуску програми, упаковуються як один образ і можуть бути використані повторно. Застосунок в контейнері працює в ізольованому середовищі і не використовує пам'ять, процесор або накопичувач пам'яті хостової операційної системи. Це гарантує ізолюваність процесів всередині контейнера.

Існують два основні варіанти віртуалізації, а точніше два підходи до створення незалежних ізольованих обчислювальних просторів на одному фізичному сервері: віртуальні машини і віртуальні контейнери. У першому випадку для кожної віртуальної машини використовується власна гостьова операційна система(ОС) і гіпервізор, а в другому – для всіх контейнерів застосовується ядро однієї хостової ОС.

Однак, оскільки віртуальні машини включають операційну систему, їх розмір може складати декілька гігабайт. Також недоліком віртуальних машин можна назвати те, що для завантаження ОС і ініціалізації програми, які в них розміщені, потрібно набагато більше часу. Контейнери займають менше простору і, в основному, їх розмір вимірюється в мегабайтах. Порівнюючи їх продуктивність з віртуальними машинами у часі, контейнери можуть запускатися майже миттєво.

Отже, основними перевагами контейнерів є:

– гнучке середовище:найбільша перевага у використанні технологій контейнерів полягає в тому, що їх можна створювати набагато швидше, ніж екземпляри віртуальних машин;

– підвищена продуктивність: контейнери збільшують ефективність розробників, за рахунок усунення міжмережових залежностей і конфліктів. Кожен контейнер може розглядатися як окремий мікросервіс.

– управління версіями дозволяє відстежувати версії контейнера, стежити за відмінностями між ними та ін.;

– портативність середовища обчислень: контейнери інкапсулюють всі відповідні деталі, такі як залежності застосунків і операційні системи, необхідні для запуску програми. Це спрощує транспортування та тиражування контейнерів з одного середовища в інше;

– безпека: контейнери ізолюють процеси одного контейнера від іншого, та від базової інфраструктури. Таким чином, будь-яке оновлення або зміна в одному контейнері не впливає на інший контейнер.

Тим не менш, контейнери мають низку недоліків, таких як:

– підтримка NativeLinux: більшість контейнерних технологій, таких як Docker, засновані на Linux-контейнерах (LXC). Тому запуск цих контейнерів в середовищі Microsoft викликає небажані труднощі;

– контейнери – відносно нова технологія на ринку. Через нестачу кваліфікованих спеціалістів, на вирішення проблем витрачається більше часу і ресурсів;

– підвищена складність: при збільшенні числа контейнерів, які працюють з додатком, також збільшується коефіцієнт складності. Управління мережею контейнерів є складним та об'ємним процесом. Такі інструменти, як Kubernetes можуть полегшити управління великою кількістю контейнерів.

Отже, проведене дослідження націлене на актуалізацію та формалізацію основних відомостей про платформу контейнеризації Docker, виділення основних схем та підходів роботи з цим застосунком, а також перспективи та подальший розвиток даної технології у системі Kubernetes, як однієї з найбільш популярних та використовуваних на сьогоднішній день.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Hello Minikube [Electronic resource]. – Access mode: <https://kubernetes.io/docs/tutorials/hello-minikube/>.

2. Docker Overview [Electronic resource]. – Access mode: <https://docs.docker.com/engine/docker-overview/>.

3. Docker vs VMWare: How Do They Stack Up? [Electronic resource]. – Access mode: <https://www.upguard.com/articles/docker-vs.-vmware-how-do-they-stack-up>.

РОЗРОБКА КРОСПЛАТФОРМНОГО МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ВІДОБРАЖЕННЯ ПОГОДИ НА FLUTTER

Перегляд погоди у смартфоні – є однією з щоденних звичок кожної людини. Саме тому розробка застосунку для комфортного перегляду прогнозу погоди є актуальною на сьогодні.

Основними вимогами до створення застосунку були:

- актуальна та повна інформація про погоду;
- можливість вибору міста для отримання погоди;
- можливість оновлення даних;
- використання на кількох програмних платформах.

Для розробки мобільного застосунку був обраний стек технологій Flutter SDK та Dart, через актуальність цієї платформи на ринку праці, а також новизни платформи, доступності та повноти навчальних матеріалів. Dart – це об'єктно-орієнтована мова, заснована на класах, із синтаксисом у стилі C. Dart може компілювати до рідного коду чи JavaScript. Він підтримує інтерфейси, комбінації, абстрактні класи, вдосконалені дженерики та умовиводи.

У комплекті з мовою програмування Dart, було використано Flutter SDK. Flutter – це набір для розробки програмного забезпечення з відкритим кодом, створений Google. Він використовується для розробки програм для Android, iOS, Windows, Mac, Linux, Google Fuchsia та Інтернету.

Під час розробки було створено програмне забезпечення, яке виконує функції показу прогнозу погоди користувачу завдяки використанню API запитів. Дані для застосунку були взяті з сайту [asuweather.com](https://api.asuweather.com) та надходили у файлі формату JSON.

Були створені класи, які використовують основні методи об'єктно-орієнтованого програмування, реалізована функція отримання даних про погоду у обраному користувачем місті за допомогою API запитів. Усі рішення для реалізації компонентів системи знаходяться в декількох файлах, об'єднаних між собою.

Застосунок для перегляду актуальної інформації щодо погоди в Україні. Він надає користувачу можливість переглянути погоду вибраного міста на день. З застосунку можна дізнатися реальну температуру, температуру, що відчувається, швидкість вітру, вологість, стан погоди та пору доби. Вибір міста відбувається у випадяючому списку. Є можливість оновлення даних за

допомоги кнопки оновлення в верхньому правому куті. Програма має зручний і простий користувацький інтерфейс.



Рисунок 1 – Зображення роботи програми

Програмні засоби, що було використано, концентруються на ідеї кросплатформності на рівні виконання. Тобто, цей застосунок для перегляду погоди може запускатись на різних програмних платформах без попередньої перекомпіляції. Як вказує розробник Flutter – компанія Google, їх платформа зберігає нативний вигляд застосунків та ідентичний вигляд на Android та IOS.

Отже, розроблений застосунок є демонстрацією поєднання досить нової мови Dart з платформою Flutter, що дало можливість для розробки кросплатформених застосунків для платформ Android та IOS одночасно.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Flutter уроки - Flutter и Dart документація [Електронний ресурс]. – Режим доступу : <https://flutter.su/docs>
2. Windmill E. Flutter in Action /Eric Windmill. – Manning Publications, 2019. – 310 p.

УДК 004.4

Мироснова Н.О.¹, Третьяков Д.А.², Сидорський В.С.², Радченко І. А.²

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² студ. гр. КНТ-228 НУ «Запорізька політехніка»

РОЗРОБКА КРОСПЛАТФОРМНОГО МОБІЛЬНОГО ЗАСТОСУНКУ «МЕНЕДЖЕР ВИТРАТ» НА FLUTTER

Мобільний застосунок – програмне забезпечення, призначене для роботи на смартфонах, планшетах та інших мобільних пристроях, та розроблено для конкретної платформи (iOS, Android, Windows Phone тощо). Завдання проекту – створити менеджер витрат, який дозволяє користувачеві спостерігати за своїми витратами та надходженнями, аналізувати їх завдяки простій статистиці, що надається застосунком.

Програма була написана за допомогою SDK Flutter на мові Dart у середовищі Microsoft Visual Studio Code. Для зберігання даних використовувалася кросплатформна реляційна система керування базами даних SQLite, схему бази даних наведено на рисунку 1.



Рисунок 1 – Схема бази даних

Застосунок має простий інтерфейс, виконаний у звичайному для операційної системи Android стилі. Користувач може створювати, видаляти, редагувати готівкові «гаманці» (рис. 2), створювати та редагувати операції, що змінюють стан «гаманця», а також зберігаються в якості історії операцій (рис. 3).

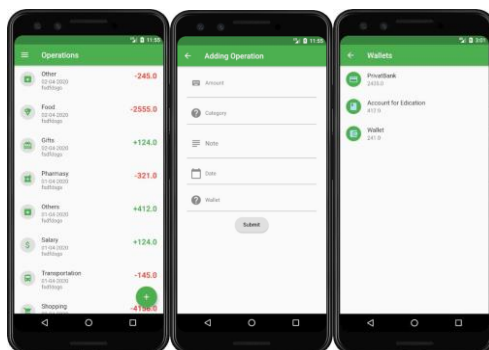


Рисунок 2 – Перелік операцій, додавання операції, вибір гаманця для операції

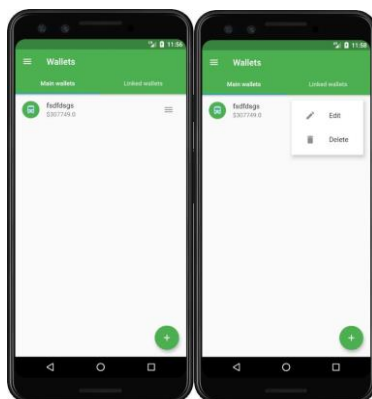


Рисунок 3 – Сторінка гаманців, Pop-Up меню зміни гаманця

Також можливо створювати «гаманці», прив'язані до банківських рахунків банку Monobank, інформація про залишок которых оновлюється за рахунок запитів до API банку. В цьому випадку операції автоматично додаються до переліку. Застосунок надає просту статистику у вигляді діаграм за категоріями витрат та сумарних значень усіх витрат та надходжень за заданий проміжок часу (рис. 4).



Рисунок 4 – Сторінка статистики

УДК 004.9

Миронова Н.О.¹, Гальченко В.В.², Перетятко В.В.², Савенко Х.С.²

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² студ. гр. КНТ-228 НУ «Запорізька політехніка»

РОЗРОБКА КРОСПЛАТФОРМНОГО МОБІЛЬНОГО ЗАСТОСУНКУ «WORLDTIME» НА FLUTTER

Flutter – це фреймворк для створення мобільних застосунків від компанії Google. Він є кросплатформним і дозволяє компілювати створений проект під операційні системи Android і iOS. Flutter активно просувається Google, поступово набирає популярність та в подальшому буде тіснити інші, які використовуються зараз засоби кросплатформної розробки.

На основі Flutter було розроблено застосунок що відображає час у різних країнах світу в залежності від їх часового поясу, використовуючи API сайту WorldTimeAPI (<http://worldtimeapi.org>).

Сайт WorldTimeAPI – простий JSON або звичайний API текст для отримання поточного часу у часовому поясі та пов'язаних із ним даних.

World Time звертається до сайту, отримує інформацію з нього та змінює її до зручного вигляду. Відбувається парсування інформації, тобто її розбиття на окремі частини (рис.1).

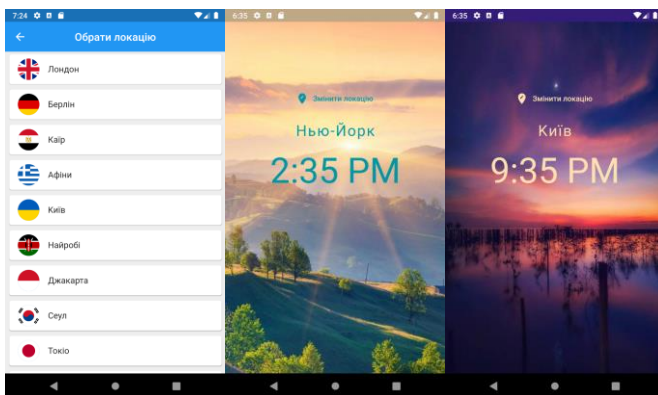


Рисунок 1 – Застосунок «World Time»

Проект має головну папку `lib`, що містить головний файл `main.dart`, папки `pages` та `services`. Папка `pages` складається з трьох сторінок:

- `choose_location.dart`;
- `home.dart`;
- `loading.dart`.

Також проект має папку `assets`, що містить у собі всі зображення що використовує застосунок:

- зображення фону у денний час;
- зображення фону у нічний час;
- зображення прапорів країн.

При запуску створеного застосунку сторінка завантажує початкове значення домашньої сторінки, а також чекає доки застосунок з'єднається через інтернет із сайтом. Використовує віджет `SpinKitFadingCircle()`, що відображає анімацію індикатору завантаження.

У `Loading` застосовано асинхронні методи програмування `await` та `async`.

Робота таких методів виводиться в окремі потоки і не порушує роботи основного потоку програми: обчислення та відображення інтерфейсу.

Застосунок містить маршрути та навігацію сторінок. Для цього застосовано клас `Navigator`, який управляє стеком віджетів, він дозволяє плавно переходити між екранами програми.

Домашня сторінка містить інформацію про час, місце, вибір локації, а також задній фон. Має віджети, що відповідають за розміщення інформації, її положення, зовнішній вид та стиль. В залежності від часу дня (день або ніч) вид заднього фону змінюється. Також є кнопка «Змінити локацію», що зроблена за допомогою віджету `FlatButton` та містить іконку зображення

edit_location.

Сторінка вибору локації показує список міст та прапори їх країн. Для відображення списку використовується віджет-клас `ListView()`. Кожен елемент списку міститься у віджеті `ListTile()`. При натисканні на елемент спрацьовує функція оновлення часу, яка змінює дані часу та міста на домашній сторінці.

Усі елементи мають зображення, які загорнуто у віджет `CircleAvatar()`, застосунок звертається до папки `assets`, використовує певне зображення та за допомогою віджету `CircleAvatar()` змінює його зовнішній вигляд.

У файлі сторінки є оголошення списку `List`, масив з інформацією про локації, кожна локація визначена як окремий елемент.

Також створена функція `updateTime`, що при виборі іншої локації оновлює дані часу та місця на головній сторінці.

УДК 004.9

Миронова Н.О.¹, Ковальов Р.В.², Резніченко А.С.², Рогова Є.В.²,
Тихоновська Т.В.²

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² студ. гр. КНТ-219сп НУ «Запорізька політехніка»

ВІЗУАЛІЗАЦІЯ КРИПТОГРАФІЧНИХ АЛГОРИТМІВ ДЛЯ BLOCKCHAIN ТЕХНОЛОГІЙ. RSA (RIVEST–SHAMIR–ADLEMAN)

Актуальність роботи полягає в практичному підході до роботи із текстом і зображенням, де введена інформація зчитується та додається до масиву, що шифрується за відповідним ключем та дешифрується за другим ключем відповідним першому повертаючи початкову інформацію.

Метою даної роботи є розробка візуалізації криптографічних алгоритмів для Blockchain технології, а саме алгоритму шифрування RSA (Rivest–Shamir–Adleman).

Задачі роботи – це створити програмний продукт, який шляхом взаємодії з користувачем шифрувати і дешифрувати зображення за створеними ключами.

Система повинна забезпечувати:

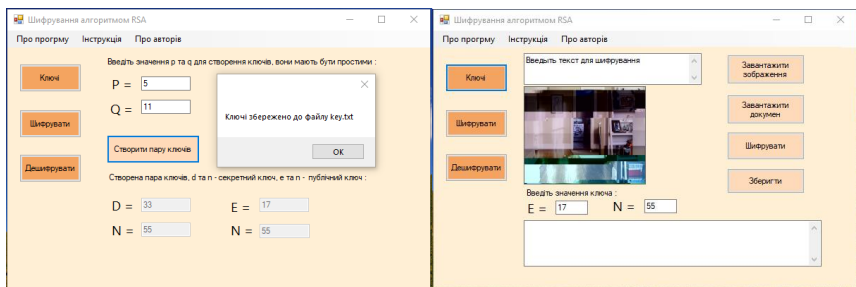
- завантаження зображення або тексту;
- зчитати зображення або текст в масив;
- шифрувати елементи масиву;
- вивести шифровану інформацію;
- завантажити шифроване зображення або текст;
- зчитати шифроване зображення або текст в масив;

- дешифрувати елементи масиву;
- вивести дешифровану інформацію.

Під час розробки було реалізовано наступні алгоритми:

- файлове зчитування та запис у файл;
- відображення візуальних та текстових елементів;
- завантаження зображення або тексту;
- розбиття зображення на байтовий масив та переведення цього масиву в інший формат;
- застосування алгоритму криптографічного шифрування RSA (Rivest–Shamir–Adleman).

На рисунку 1 представлено результати виконання розробленого коду.



а) Генерація ключів;

б) Шифрування зображення

Рисунок 1 – Приклади екранних форм програми

Під час виконання роботи було розроблено імітаційну програму на тему «Візуалізація криптографічних алгоритмів для Blockchain технології. RSA (Rivest–Shamir–Adleman)» для роботи з даними для забезпечення відображення алгоритму шифрування і дешифрування.

Створена імітаційна програма організовує систему для забезпечення вибору даних: зображення, текстовий файл. Для шифрування та дешифрування було використано криптографічний алгоритм RSA (Rivest–Shamir–Adleman). Таким чином, розробивши імітаційну програму можна зробити висновок, що алгоритм є ефективним для шифрування даних, проте його час роботи залежить від розміру цих даних. Відповідно, якщо файл є великим тоді і час виконання є довшим.

Вирішені наступні завдання роботи:

- виконано огляд сучасних програмних засобів реалізації візуалізації;
- розглянуті можливості та переваги мови програмування C#;
- розглянуті можливості та переваги середовища розробки JetBrains Rider;

– розроблено систему для відображення результатів роботи алгоритму шифрування RSA.

РОЗРОБКА SPA ЗАСТОСУНКІВ ЗА ДОПОМОГОЮ JAVASCRIPT ФРЕЙМВОРКУ VUE.JS

По перше, розглянемо різницю між поняттям «фреймворк» та «бібліотека» для того щоб зрозуміти чому Vue.js вважається фреймворком. Отже, фреймворк – це програмне середовище спеціального призначення, своєрідний каркас, який використовується для того, щоб істотно полегшити процес об'єднання певних компонентів при створенні програм. Це основа, яка дозволяє додавати компоненти в залежності від потреб. База, на якій можна сформувавши програму будь-якого призначення досить швидко і без особливих труднощів. Бібліотека – це збірник класів або функцій на мові JavaScript. Тож Vue – це прогресивний фреймворк для створення користувацьких інтерфейсів. На відміну від фреймворків-монолітів, Vue створений придатним для поступового впровадження. Його ядро в першу чергу вирішує завдання рівня уявлення (view), що спрощує інтеграцію з іншими бібліотеками та існуючими проектами. З іншого боку, Vue повністю підходить і для створення складних односторінкових застосунків (SPA, Single-Page Applications), якщо використовувати його спільно з сучасними інструментами та додатковими бібліотеками.

Творцем Vue.js є Еван Ю, колишній співробітник Google. Почав він розробляти фреймворк в 2013 році, а в лютому 2014 року відбувся перший публічний реліз. Vue широко використовується серед китайських компаній, наприклад: Alibaba, Xiaomi. Він входить в ядро Laravel і PageKit. Нещодавно вільна система управління репозиторіями GitLab теж перейшла на Vue.js. В кінці вересня 2016 вийшов в реліз Vue.js 2.0, з більш кращими оновленнями і з упором на продуктивність – тепер використовується віртуальний DOM, підтримується серверний рендеринг, можливість використовувати JSX та ін. Хоча зараз він підтримується тільки співтовариством, він тримається гідно навіть на рівні продуктів таких гігантів, як Google і Facebook (Angular2 і React 15), і поступово наздоганяє їх по популярності.

SPA (single page application) – це тип веб застосунків, в яких завантаження необхідного коду відбувається на одну сторінку, що дозволяє заощадити час на повторне завантаження одних і тих же елементів.

Зрозуміти, наскільки зручними і корисними для користувачів бувають односторінкові додатки або SPA, можна на прикладі кількох популярних сервісів гіганта Google: Gmail та Google Translate. Користувачі постійно

використовують дані сервіси, і навряд чи у кого-то виникає бажання перейти на десктопні аналоги.

Робота SPA додатку полягає у з'єднанні клієнтської частини з REST API. Клієнт посилає запит на сервер, сервер повертає дані у форматі JSON, клієнт за допомогою певних скриптів виводить дані у потрібному для додатка вигляді.

Для побудови frontend частини SPA застосунку на Vue.js необхідні такі інструменти: пакетний менеджер NPM, за допомогою якого можна встановлювати всі необхідні для розробки пакети, Vue-cli – рекомендований офіційний шаблон проєктів на Vue.js з налаштуваннями Webpack. За допомогою команди `npm install -g vue-cli` встановлюємо Vue-cli, після цього виконуємо команду `vue init vue-spa`, таким чином в нас з'явиться готовий шаблон проєкту який нам презентував vue-cli. Для розмежування бізнес-логіки програми та користувацького інтерфейсу нам знадобиться інструмент Vuex (`npm i --save-dev vuex`). Це дуже зручно, коли логіка (код) програми відділена від html-структури. Для зв'язку з REST API знадобиться бібліотка AXIOS (`npm i --save-dev axios`). У ролі REST API щоб не витратити час на його розробку можна використати такий сторонній сервіс (псевдо REST API) Firebase від Google, або `jsonplaceholder.typicode.com`.

Структура проєкту. Що ж містить шаблон проєкту vue-cli? По перше це всі необхідні залежності проєкту, (файл `package.json`), файл з налаштуваннями Webpack, файл `index.html` містить просту розмітку HTML з єдиним елементом "app" в body. Він буде замінений на DOM, згенерований vue. З цієї причини тег body не рекомендується використовувати в якості кореневого елемента, також файл `gitignore` та `readme` – для роботи з віддаленим git-репозиторієм. Також в структурі можна побачити папку `src`, в якій зберігається точка входу проєкту (файл `main.js`), папку `components` (збереження окремих компонентів vue), папку `router` (тут зберігаються маршрути окремих компонентів), файл `App.vue` в якій зберігається основна частина додатку, та папка `assets` (для зображень або файлів стилю), особисто для роботи з бізнес-логікою проєкту створено папку `store`, де зберігаються файли які містять всі необхідні об'єкти Vuex (`store`, `actions`, `mutations`, `getters`).

Vue-component. Кожен файл `*.vue` складається з блоків трьох типів: `<template>`, `<script>` і опціонально `<style>`.

В результаті, можна розділити проєкт на пов'язані компоненти. Усередині компонента його шаблон, логіка і стилі невід'ємно пов'язані, і їх поєднання фактично робить компонент більш цілісним і легко підтримуваним.

ПРОГРАМНА РЕАЛІЗАЦІЯ ВІЗУАЛІЗАЦІЇ АЛГОРИТМІВ СОРТУВАННЯ

На сьогодні наш світ неможливо уявити, наприклад, без телебачення або інтернету. Фізіологічно людина влаштована так, що зорове сприйняття є основним каналом отримання знань. Завдяки зору ми сприймаємо близько 90% інформації. За результатами досліджень встановлено, що продуктивність праці людини, яка використовує візуальну інформацію, вища, ніж з використанням, наприклад, текстової інформації.

Візуалізація має критичне значення для аналізу даних. Вона допомагає нам розкрити складну внутрішню структуру даних, яку неможливо усвідомити в інший спосіб. Дані – послідовність бітів та байтів, що зберігається у файлі на жорсткому диску комп'ютера – є невидимими. Щоб побачити їх та зрозуміти їхнє значення, нам потрібно візуалізувати ці дані. Саме для цього існують різноманітні програмні засоби для візуалізації інформації.

Критичним моментом даних засобів є їх нестача і, як наслідок, низька якість цих продуктів. Програми демонструють лише візуалізацію сортування одновимірних масивів даних з малою кількістю налаштувань. З цього випливає, що користувач не має можливості побачити візуалізацію сортування матриці елементів, тобто двовимірного масиву. Створення програми, яка вміє це робити, дало можливість вивчати сортування даних в такому поданні.

Розроблена програма має бути простою, доступною і в той же час надавати можливість певного роду кастомізації налаштувань за власним бажанням. Саме через простоту і поширення було обрано інтерфейс програмування додатків – Windows Forms. Це дозволить дуже легко додавати нові функції, зберегти простоту та, використовуючи можливості останніх версій Microsoft .NET Framework, застосовувати сучасні засоби розробки. Інтерфейс програми має приваблювати користувача, зберігати просту і прозору логіку. Для початку візуалізації сортування необхідно виконати лише одну дію – натиснути кнопку «Почати».

Візуалізація алгоритмів сортування в програмі влучно показує їхню роботу, завдяки простій та водночас ефективній реалізації. Перший крок на шляху візуалізації – це створення об'єкту BitMap, який в свою чергу є класом для роботи з растровими зображеннями. Другим кроком є відкриття картинки

за допомогою новоствореного об'єкта. Далі переходимо до сортування матриці значень зеленого кольору пікселів і в момент, коли відбувається зміна елементів, на зображенні ці елементи виділяються червоним кольором. Далі виводимо картинку на відповідне місце на головній формі програми, переставляються пікселі зі значеннями змінюваних елементів і знову оновлюється картинка. Кроки виділення і перестановки пікселів відбувається до кінця сортування зображення.

Програма буде використовувати параметри, встановлені користувачем у відповідних елементах форм. Для простоти використання програми було створено 5 форм, які взаємодіють між собою(рис.1).

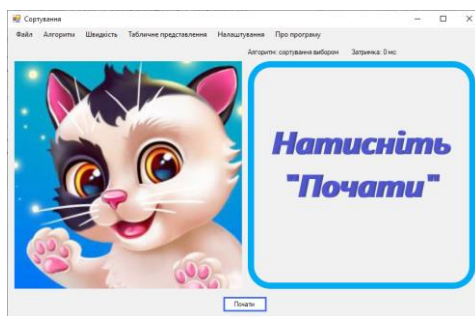


Рисунок 1 – Головне вікно програми

За роботу з BMP-зображеннями буде відповідати бібліотека `bmp_image`. Файли типу *.csv та *.json використані для збереження значень кольорів пікселів до та після сортування. Клас `MyException` повідомляє користувача про відповідні помилки, що можуть виникнути в процесі виконання програми. Візуалізація відбувається завдяки алгоритмам сортування, а саме `Select sort` та `Insertion sort`. Послідовність виконання сортування та його швидкість залежить від алгоритму (рис.1).

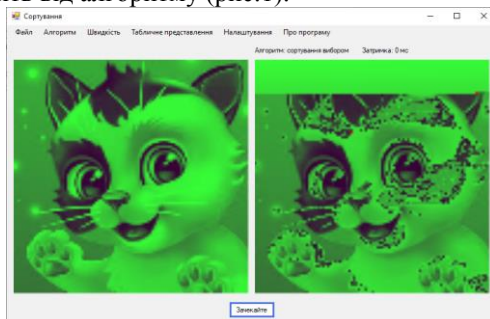


Рисунок 2 – Процес візуалізації

В результаті створення даної програми була виконана робота з алгоритмами сортування, продемонстровано процес візуалізації алгоритмів сортування на прикладі ВМР-зображення, показані відмінності їх алгоритмів, виконана робота з класами, сторонньою бібліотекою, розроблено графічний інтерфейс.

УДК 004.9

Миронова Н.О.¹, Гальченко В.В.²

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² студ. гр. КНТ-228 НУ «Запорізька політехніка»

ВІЗУАЛІЗАЦІЯ АЛГОРИТМІВ СОРТУВАННЯ

Алгоритми сортування упорядковують елементи за певним критерієм. Візуалізація алгоритмів сортування служить для того, щоб зрозуміти, як працюють одні з найвідоміших алгоритмів сортування. Таким чином, для реалізації було обрано два алгоритми: сортування бульбашкою (Bubble sort) та порозрядне сортування (Radix sort).

Програма була написана на мові C++ та у середовищі програмування Microsoft Visual Studio. Для графічного представлення алгоритмів було використано Windows Forms.

Програма поділена на декілька етапів, в одному з яких і застосовуються обрані алгоритми. Для початку роботи потрібно натиснути на кнопку "Відкрити зображення" та обрати потрібне зображення (рис.1). Далі, натиснувши на кнопки "Операції з зображенням", "Розбиття", пікселі перемішаються. Після рандомізування зображення натиснути на кнопку "Відновлення", де можна обрати алгоритм сортування, який Вам до вподоби та процес візуалізації запускається. Також у програмі було реалізовано наступні форми: "Про авторів", "Про програму", "Інструкція", "Статистика". Ці форми можна знайти натиснувши на кнопку "Довідка".

Алгоритм сортування Bubble sort. Сортування бульбашкою або сортування простими обмінами – один з найпростіших алгоритмів сортування. Він застосовується для упорядкування масивів невеликих розмірів.

Суть алгоритму в тому, що відбувається кілька проходів по масиву. При кожному проході попарно порівнюються два сусідні елементи. Якщо вони знаходяться в правильному порядку, то нічого не відбувається, в іншому випадку вони міняються місцями. В результаті першого проходу максимальний елемент виявиться в кінці, тобто спливе немов бульбашка. Потім все повторюється до того моменту поки весь масив не буде

відсортований.

Хоч алгоритм бульбашкою є простим в реалізації, проте сам процес роботи сортування проходить дуже повільно. Тож для роботи з великими за обсягом зображеннями алгоритм бульбашка не підходить (рис. 1.2).

Алгоритм сортування Radix sort. Сортування за розрядами – сортування за розрядами. Існує два різновиди: LSD (least significant digit) і MSD (least significant digit). У першому випадку відбувається сортування елементів за молодшим розрядам (все що закінчується на 0, потім на 1 і так до 9). Після цього вони групуються за наступним з кінця розряду, поки вони не закінчаться. У MSD сортування відбувається по старшому розряду.

У порівнянні з алгоритмом бульбашка, алгоритм порозрядного сортування є складним у реалізації, однак процес сортування є швидким. Цей алгоритм підходить як для маленьких зображень, так і для великих.

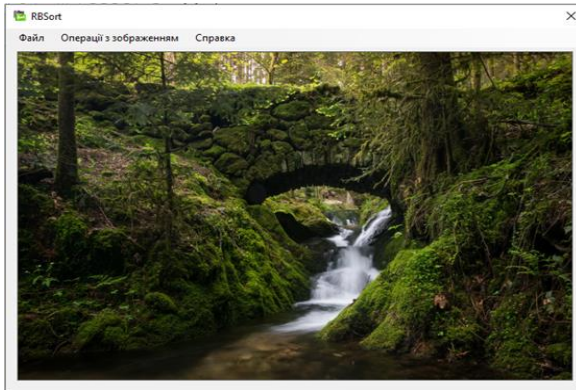


Рисунок 1 – Головна форма програми

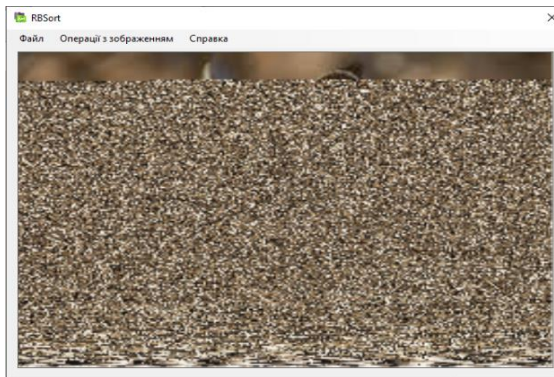


Рисунок 2 – Алгоритм сортування Bubble sort

РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ПЕРЕГЛЯДУ ТА АНАЛІЗУ ДАНИХ МАТЧІВ ТА СТАТИСТИКИ ГРАВЦІВ БАГАТОКОРИСТУВАЦЬКОЇ ГРИ

Відеоігри вже давно перестали бути лише засобом відпочинку. Більшість багатокористувацьких ігор мають змагальну складову, наприклад, глобальних рейтинг гравців, турніри, тощо, а кіберспорт у деяких країнах не поступається у популярності традиційному.

Ключовим фактором для підвищення рівня ігрових навичок є збір та аналіз інформації. Професійний кіберспорт з'явився у кінці XX століття, і методи збору й аналізу інформації були не дуже ефективними. Завдяки сучасному вбудованому інструментарію для аналізу даних ігрових сесій, середній рівень гравців значно виріс, адже завдяки швидкому та зручному аналізу інформації гравець може зрозуміти, на які аспекти своєї гри йому треба звернути увагу, або обрати найефективніші шляхи для здобуття перемог.

Розроблена платформа має надавати гравцеві свіжу інформацію щодо ігрових класів у випадку змін до ігрового процесу, бо всі змагальні ігри завжди прогресують, і змінюються як окремі правила гри, так і весь формат змагань. Також інформація має надаватися у зручному для сприйняття вигляді, адже ніхто не схоче витратити більшість часу на дешифрацію сирих даних. Інтерфейс має бути максимально простим і логічним, а навігація повинна спрощувати пошук потрібної інформації. Попередній вигляд екрану зі статистикою матчу зображений на рисунку 1.

Так як розроблена платформа має бути доступною для більшості гравців і середній сеанс є нетривалим, застосунок розроблявся для мобільних пристроїв. Це обумовлено тим, що у багатьох країнах розповсюдженість смартфонів значно перевищує даних параметр у персональних комп'ютерів, так як комп'ютери потрібні лише для роботи, а для ігрового процесу використовуються ігрові консолі. Також володарі смартфонів значно більше схильні до спонтанних дій, адже, на відміну від персонального комп'ютера, смартфон є підручним пристроєм, і запуск будь-якого застосунку займає набагато менше часу. Це забезпечить більший попит для застосунку, так як користуватись ним можуть навіть люди з невеликою кількістю вільного часу.

Клієнтська частина буде використовувати апаратні засоби системи для побудови діаграм. Для обміну даними з сервером, на якому знаходиться база

даних, використовується PHP-скрипт, що приймає запит і відсилає його на сервер. Користувачеві повертається JSON-файл, який дешифрується вже на стороні клієнта. За рендер усіх діаграм та графіків відповідатиме бібліотека з відкритим кодом – MPAndroidChart.

Серверна частина розроблена за допомогою СУБД PostgreSQL.

Клієнтська частина була розроблена у середовищі розробки Android Studio, програмний код був виконаний мовою Kotlin. Вибір мови програмування був зумовлений тим, що нещодавно саме ця мова стала основною для розробки на ОС Android, витіснивши застарілу мову Java. Зараз мова Kotlin першою отримує найсвіжіший інструментарій та бібліотеки для вирішення проблем.



Рисунок 1 – Попередній вигляд екрану зі статистикою матчу

РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ПЛАНУВАННЯ ТА ОБЛІКУ ЧАСУ, ВИТРАТ ТА ЗАВДАНЬ

Проблема збереження часу та підвищення ефективності людини завжди є актуальною. Бо чим більше часу тим більше людина зможе виконати задач, чим більше виконаних задач тим більша ефективність, а чим більше ефективність тим більша користь яку приносить людина собі або компанії працівником якої вона є.

З появою мобільних пристроїв та на даний час, їх масовим розповсюдженням та користуванням, майже кожна людина зараз має у своєму кармані цей міні комп'ютер, і тепер ви можете виконати більшість завдань, для яких комп'ютер був необхідним, будь-коли та будь-де, навіть коли йдете по вулиці чи просто не маєте можливості скористатися ноутбуком, і це відніме у вас лише мить.

Ринок застосунків для мобільних пристроїв ще відносно молодий, та ніша програм для продуктивності вже налічує у собі сотні, якщо не тисячі, «To do» застосунків або «тайм трекерів», проте майже неможливо знайти програму яка б об'єднувала кращі риси цих застосунків, яка б давала можливість не закриваючи його створити собі завдань на майбутній тиждень і одночасно розібрати минулі задачі або таймери що враховували ваш робочий час, швидко перемістивши їх до звіту. Враховуючи це нагальною і є необхідність створення застосунку, що містить вище наведені функції.

Такий застосунок став би набагато зручнішою альтернативою багатьом програмам, об'єднавши кращі їх можливості та функції, і забезпечивши легку та зрозумілу взаємодію між ними. Він заощадив би багато часу для кожної людини, яка зацікавлена у керуванні своїми завданнями, часом та витратами.

Саме така мета лежить в основі створюваного додатку. Зробити застосунок який би мав у собі можливості відстежити час витрачений на певні задачі, які ви самі собі занесли до календаря і встановили дедлайни, відзначили по яким проектам є ці задачі, та встановили нагадування, щоб не забути про них, на основі таймерів та існуючих задач створити звіт який буде містити всю необхідну інформацію та при цьому його можна буде заповнити кількома рухами пальця.

Розроблений додаток має бути інтуїтивно зрозумілим, легким у використанні та швидким. Він має надати можливість керувати та планувати свій час, завдання та витрати, при цьому забезпечивши повну взаємодію цих

частин застосунку одна з одною для отримання максимально приємного досвіду використання та пришвидшення завдань які необхідно було б робити вручну або переносити з одного застосунку до іншого. Також він повинен забезпечити роботу як з підключенням до мережі інтернет, так і без нього.

Програма створена за допомогою середовища розробки XCode, мовами Objective C та Swift, для пристроїв під керуванням операційної системи iOS версії 11.0 та вище.

Приклад інтерфейсу представлений вікном керування та планування завданнями наведено на рисунку 1.

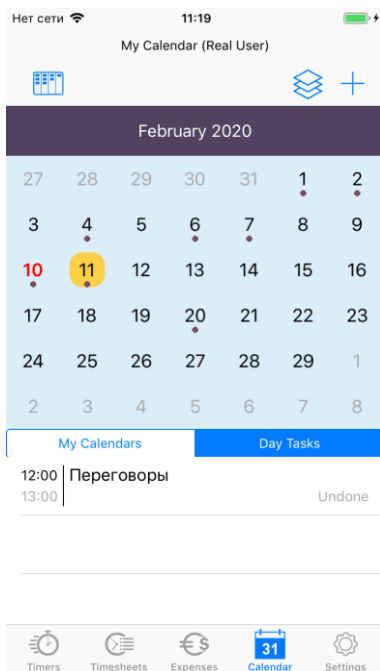


Рисунок 1 – Вікно керування завданнями

Під час розробки були використані сторонні бібліотеки: FirebaseAnalytics – для збору аналітики використання, FirebaseCrashlytics – для збору даних про збої застосунку, GoogleApiClientForREST – для синхронізації завдань календаря застосунку з Google календарем користувача, GoogleSignIn – для входу до акаунту Google, ReachabilitySwift – для перевірки підключення до інтернету та реагування на зміну підключення, SBJson4 – для обробки json.

За збереження даних офлайн відповідає вбудована база даних SQLite та бібліотека CoreData, а за збереження онлайн – PostgreSQL. Зв'язок між базою даних PostgreSQL та застосунком забезпечуть скрипти PHP.

УДК 004.9

Миронова Н.О.¹, Куделя Д.С.²

¹ канд. техн. наук, НУ «Запорізька політехніка»

² студ. гр. КНТ-147сп НУ «Запорізька політехніка»

РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ МАТЧМЕЙКІНГУ ДЛЯ БАГАТОКОРИСТУВАЦЬКИХ ОНЛАЙН ВІДЕОІГОР

Технічний прогрес у сфері обчислювальної техніки і розвиток інформаційних технологій не обійшов стороною індустрію відеоігор. Сучасні комп'ютерні ігри – одні з найпопулярніших програм на комп'ютері.

Найбільший обсяг аудиторії на сьогоднішній день мають саме мережеві багатокористувацькі сесійні ігри. Ігрові сесії займають зазвичай невелику кількість часу та мають чіткі часові межі, тому вони досягли значної популярності, адже контролювати ігровий час стало занадто легше.

Інтернет-сесії багатокористувацьких ігор можна організувати кількома способами. Матчмейкінг є варіантом, який дає гравцям можливість потрапити в гру з меншими зусиллями. Гравці по одному або у командах шукають гру, і система з'єднує з іншими подібними гравцями. Після того, як знайдено відповідну кількість гравців, матч створюється і гра може розпочатися.

Матчмейкінг (matchmaking) у відеоіграх – процес об'єднання гравців за певними параметрами у спільні сесії онлайн-гри.

Багато систем матчмейкінгу використовують оцінки навичок гравця або ранги. Це дозволяє гравцям подібної здатності грати один проти одного, даючи більш близькі і більш конкурентоспроможні ігри для всіх діапазонів навичок.

Система складається з наступних розроблюваних модулів:

- клієнт матчмейкінгу, що інтегрується в ігровий клієнт – відповідає за з'єднання з доступним сервером матчмейкінгу та надання інформації ігровому клієнту щодо знайденого ігрового сервера;

- сервер матчмейкінгу – відповідає за отримання ігрової інформації, її сортування за відповідними параметрами, з'єднання з доступним сервером підтримки ігрових серверів для запиту розгортання ігрового сервера, відсилання інформації про ігровий сервер клієнтам матчмейкінгу, для котрих було знайдено сходження, при власному розгортанні – надає інформацію про

готовність до бази даних доступних серверів матчмейкінгу;

– сервер підтримки ігрових серверів (сервер розгортання) – відповідає за розгортання та працездатність ігрових серверів, при власному розгортанні – надає інформацію про готовність до бази даних доступних серверів підтримки;

– база даних доступних серверів матчмейкінгу – зберігає інформацію про доступні в поточний час сервери матчмейкінгу;

– база даних доступних серверів підтримки ігрових серверів – зберігає інформацію про доступні в поточний час сервери підтримки ігрових серверів.

Структурна схема системи матчмейкінгу зображена на рисунку 1.

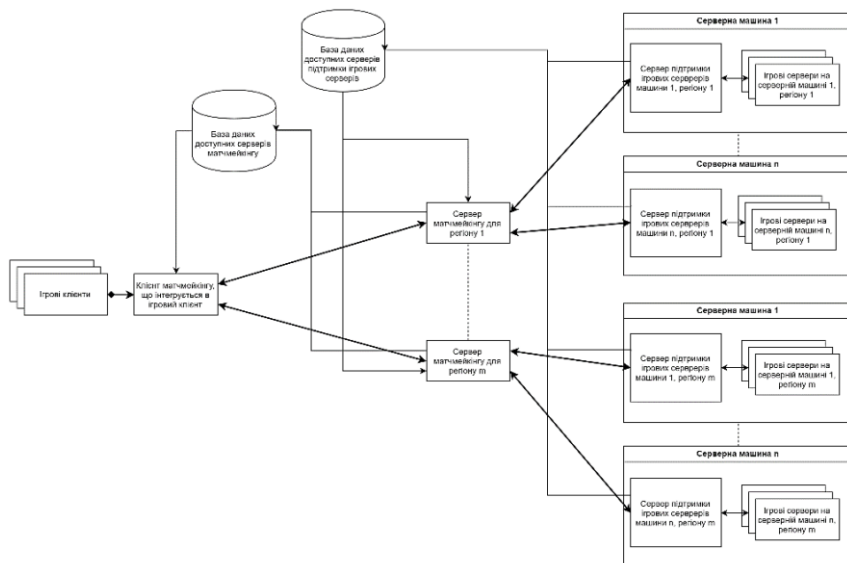


Рисунок 1 – Структура програмної системи

Схема взаємодії модулів системи між собою зазначена на рисунку 2.

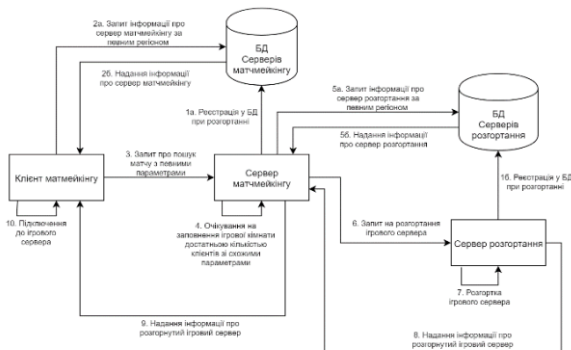


Рисунок 2 – Взаємодія модулів системи між собою

Розроблена система здійснює підбір гравців із схожими внутрішньоігровими характеристиками за певним регіоном, розгортає ігрові сервери для підібраних гравців, та підключає гравців до ігрових серверів.

Система має високу здібність до масштабованості, що дозволяє задовольнити потреби проєктів будь-яких розмірів.

УДК 004.9

Миронова Н.О.¹, Колпакова Д.В.²

¹ канд. техн. наук, НУ «Запорізька політехніка»

² студ. гр. КНТ-126 НУ «Запорізька політехніка»

РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ІНВЕНТАРИЗАЦІЇ ТОВАРНО-МАТЕРІАЛЬНИХ ЦІННОСТЕЙ

Розкрадання товарно-матеріальних цінностей (ТМЦ) – це проблема, з якою стикаються багато компаній, що ведуть свою діяльність в різних галузях вітчизняної економіки. Для вирішення цієї проблеми запропоновано створення мобільного застосунку для автоматизованої інвентаризації майна підприємства з використанням QR-кодів.

За допомогою фреймворку Flutter було розроблено мобільний кросплатформний застосунок для обліку майна компанії, який працює з трьома видами користувачів: Користувач, Адміністратор, Супер Адміністратор. Приклади роботи з застосунком наведено на рисунку 1.

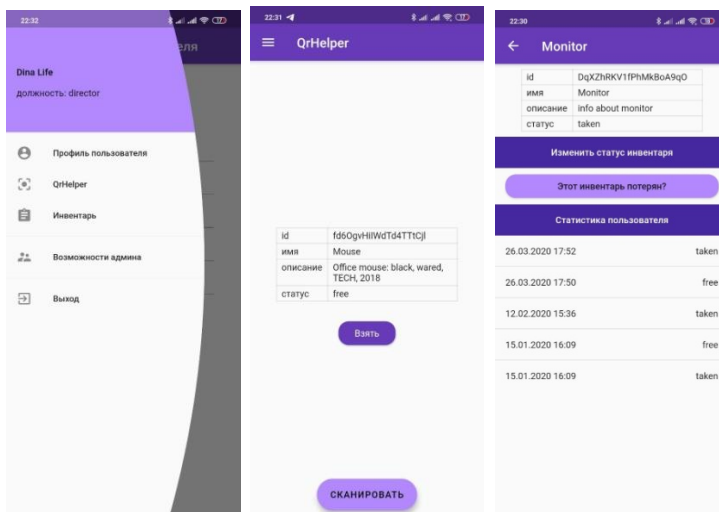


Рисунок 1 – Застосунок «QR Helper»

Користувач може сканувати QR-коди інвентарю та виконувати необхідні дії з отриманими даними.

Адміністратор має можливість переглянути весь список користувачів, список інвентарю, додати новий інвентар або видалити існуючий.

Супер Адміністратор зможе додавати до списку адміністраторів користувачів або видаляти з цього списку адміністраторів.

Даний програмний продукт використовує CloudFirestore. Слід звернути увагу, що база даних CloudFirestore не дуже зручна та часто використовується для демо версій. Також Firestore залежний від платформи, тому необхідно втручатися у нативний код та робити налаштування для кожної платформи згідно з документацією Firestore. Наприклад, для модулю Android потрібно додати google-services.json для коректної роботи застосунку. Саме тому частина, яка відповідальна за роботу з базою даних, повинна залежати від абстрактного класу або інтерфейсу.

Для розробки було використано своєрідний архітектурний шаблон похідний від MVP, тобто адаптивний шаблон, де класи з суфіксом Page виконують функцію представлення, з суфіксом Repo – постачальника даних, з Presenter – шару між Page та Repo. Сам Presenter не знає про те, який клас, що відповідний за представлення, його використовує, але знає про інтерфейс або інтерфейси репозиторіїв які постачають дані зі серверу. Діаграму класів проекту, яка відображає шаблон похідний від MVP, зображено на рисунку 2.

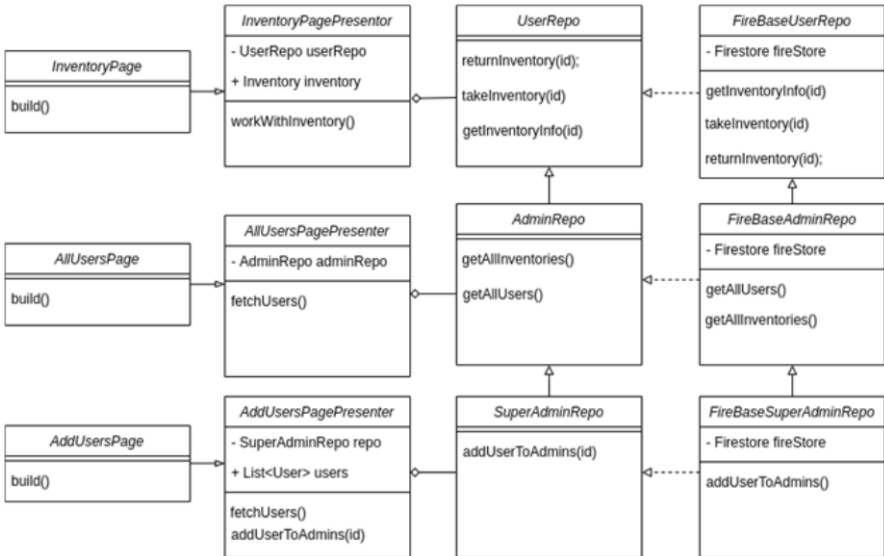


Рисунок 2 – Діаграма класів проєкту

Весь проєкт має залежність тільки від абстрактних класів UserRepo, AdminRepo та SuperAdminRepo, тому похідні від них класи можна замінити на будь яку реалізацію їх інтерфейсів. У даному випадку це FirebaseUserRepo, FirebaseAdminRepo та FirebaseSuperAdminRepo. Слід зауважити, що доступ до цих класів-репозиторіїв здійснюється за принципом Dependency Injection. У головному методі проєкту – main, з якого починається робота будь-якої програми на мові Dart, знаходиться єдина згадка про реалізовані класи. Напрямую, в програмному коді, вони не використовуються.

Підводячи підсумок, код розробленого застосунку написано з можливістю подальшого масштабування та змін шляхом використання MVP архітектури та принципів SOLID.

РОЗРОБКА ГРИ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ UNITY ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ

Мобільна гра – ігрова програма для мобільних пристроїв, наприклад стільникових телефонів, смартфонів, комунікаторів, КПК та інших (за винятком ноутбуків).

На сьогодні найбільш популярними є операційні системи Android та IOS. Ігровий рушій дозволяє створювати мобільні застосунки для обох цих платформ, а також є можливість створення багатоплатформних ігрових застосунків.

Android – операційна система і платформа для мобільних телефонів та планшетних комп'ютерів, створена компанією Google на базі ядра Linux. Підтримується альянсом Open Handset Alliance.

Android базується на ядрі Linux. Базовим елементом цієї операційної системи є реалізація Dalvik віртуальної машини Java, і все програмне забезпечення і застосування спираються на цю реалізацію Java.

Unity – багатоплатформовий інструмент для розробки двовимірних та тривимірних додатків та ігор, що працює на операційних системах Windows і OS X. Створені за допомогою Unity застосунки працюють під системами Windows, OS X, Android, Apple iOS, Linux, а також на гральних консолях Wii, PlayStation 3 і Xbox 360.

Редактор Unity має простий Drag & Drop інтерфейс, який легко налаштовувати, що складається з різних вікон, завдяки чому можна проводити налагодження гри прямо в редакторі. Рушій підтримує три сценарних мови: C #, JavaScript (модифікація). Проект в Unity ділиться на сцени (рівні) – окремі файли, що містять свої ігрові світи зі своїм набором об'єктів, сценаріїв, і налаштувань. Сцени можуть містити в собі як, об'єкти (моделі), так і порожні ігрові об'єкти – тобто ті, які не мають моделі. Об'єкти, в свою чергу містять набори компонентів, з якими і взаємодіють скрипти. Також у них є назва (в Unity допускається наявність двох і більше об'єктів з однаковими назвами), може бути тег (мітка) і шар, на якому він повинен відображатися. Так, у будь-якого предмета на сцені обов'язково присутній компонент Transform – він зберігає в собі координати місця розташування, повороту і розмірів по всіх трьох осях. У об'єктів з видимою геометрією також за умовчанням присутній компонент Mesh Renderer, що робить модель видимою.

C# – об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Переїнявши багато що від своїх попередників – мов C++, Object Pascal, Модуля і Smalltalk – C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад множинне спадкування класів (на відміну від C++).

Також Unity підтримує фізику твердих тіл і тканини, фізику типу Ragdoll. У редакторі є система успадкування об'єктів; дочірні об'єкти будуть повторювати всі зміни позиції, повороту і масштабу батьківського об'єкта. Скрипти в редакторі прикріплюються до об'єктів у вигляді окремих компонентів.

Розроблено мобільну гру (рис. 1). Ігровий застосунок було розроблено для смартфонів під керуванням операційної системи Android. У подальшому планується портування гри на смартфони під керуванням операційної системи IOS.



Рисунок 1 – Реалізація мобільної гри у ігровому рушію Unity

ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСОБІВ СИНТАКСИЧНОГО АНАЛІЗУ ЗОВНІШНІХ ДАНИХ

На сьогодні збір даних стає більш складним, і ємним процесом. З цим зіштовхнулися майже всі, користувачі інтернету. Це не завжди пошук реферату чи якоїсь інформації. Багато хто, хоче купити щось дешевше ніж у магазині та на допомогу приходять різноманітні сервіси для пошуку товарів з порівнянням цін. Цим займаються великі сервіси, які використовують синтаксичний аналіз усіх веб-сайтів, які можуть продавати цей товар. Найбільш активно парсинг використовується у пошукових системах, таких як: Google, Yahoo, Bing тощо. Пошукова система, по запиту користувача шукає потрібну йому інформацію. Також парсинг використовують для швидкого заповнення товарів для інтернет магазинів, чи для аналізу ринку або ж великого та середнього бізнесу.

Тема синтаксичного аналізу чи парсингу існує давно, але набирати свою популярність почала нещодавно. Парсинг дозволяє за лічені хвилини зібрати велику кількість інформації, чи проаналізувати щось на плагіат, помилку в синтаксисі програми, чи в орфографічному правописі. Тобто це набір певних правил, які повинні виконуватись за для збору та аналізу даних.

Алгоритм парсеру потребує швидкої обробки інформації за певними правилами, тобто регулярні вирази. Щоб парсер розумів регулярні вирази, він повинен бути написаний, на мові, що підтримує роботу з рядками. Така можливість є у Php, Ruby тощо. Регулярні вирази, описуються синтаксисом UNIX, який хоч і вважається застарілим, але широко використовується завдяки властивості зворотній сумісності.

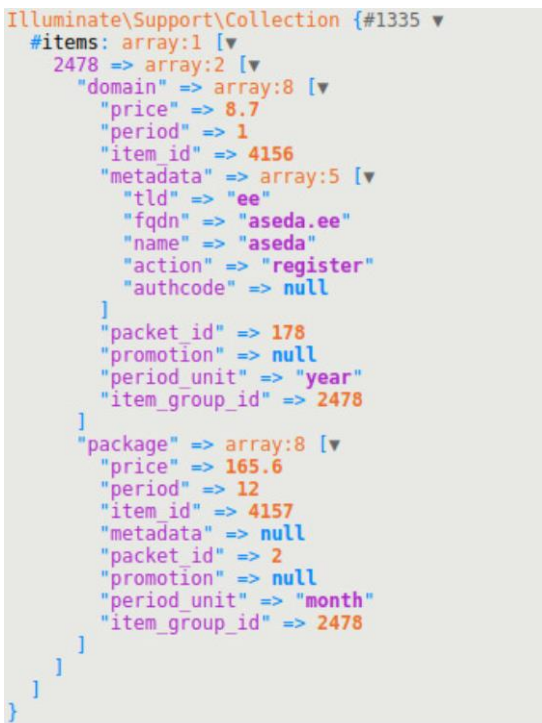
Для розробки парсеру необхідно обрати певну структуру даних. В нашому випадку це інтернет-всесвіт, який будується за певною структурою. Ця структура написана мовою HTML(HyperText Markup Language). Структура схожа на XML, та є похідною від неї. Парсинг цієї структури, виконується по відкритому та закритому тегу <назва тегу>. Дані, які містяться у тегу </назва тегу>. По цьому можна аналізувати та збирати необхідні дані.

В роботі необхідно було виконати збір даних з сервісу перегляду серіалів за такими складовими: назва, жанр, рік випуску, статус серіалу.

Отже, був виконаний аналіз сервісу для перегляду серіалів, тобто його DOM дерево. Також було проаналізовано внутрішні запити цього сервісу для

більш детального аналізу та створення правил.

Для написання алгоритму було використано мову програмування php та бібліотеку для роботи з DOM деревом. Після парсингу, усе було об'єднано в зручні масиви. (рис. 1.)



```
Illuminate\Support\Collection {#1335 ▼
  #items: array:1 [▼
    2478 => array:2 [▼
      "domain" => array:8 [▼
        "price" => 8.7
        "period" => 1
        "item id" => 4156
        "metadata" => array:5 [▼
          "tld" => "ee"
          "fqdn" => "aseda.ee"
          "name" => "aseda"
          "action" => "register"
          "authcode" => null
        ]
        "packet id" => 178
        "promotion" => null
        "period_unit" => "year"
        "item_group_id" => 2478
      ]
      "package" => array:8 [▼
        "price" => 165.6
        "period" => 12
        "item id" => 4157
        "metadata" => null
        "packet id" => 2
        "promotion" => null
        "period_unit" => "month"
        "item_group_id" => 2478
      ]
    ]
  ]
}
```

Рисунок 1 – Дані розсортовані по масивам

Отже, для написання такого парсеру дозволяє спростити та автоматизувати пошук потрібної інформації.

РОЗРОБКА МОБІЛЬНОЇ КАРТИ ПУНКТИВ БЕЗКОШТОВНОЇ ВИДАЧІ ВОДИ

Насьогодні є актуальним використання мобільних карт для представлення потрібної інформації користувачеві. Карта це візуальне подання інформації. Найбільш відомим програмним продуктом є Google Maps. Карти Google або Google Maps – безкоштовний картографічний сервіс від компанії Google, а також набір застосунків, побудованих на основі цього сервісу й інших технологій Google. У жовтні 2005 року компанія Google представила застосунок на Java під назвою «Google Maps for Mobile», призначений для роботи на будь-якому телефоні або мобільному пристрої на базі Java.

Задачею роботи було розробити мобільний застосунок, що представляє собою карту пунктів безкоштовної видачі води. Основні функції мобільного застосунку наступні:

- авторизація користувача;
- реєстрація користувача;
- відновлення паролю;
- завантаження карти, на якій зображено пункти безкоштовної видачі води (для відображення карти використовувався сервіс Google Maps);
- помітка та перевірка нової локації пункту безкоштовної видачі води;
- редагування існуючих локацій в адміністративній панелі;
- україномовний та англійськомовний інтерфейс мобільного застосунку.

При розробці мобільного застосунку використовувався наступний архітектурний шаблон проектування MVP або Model View Presenter. MVP – шаблон проектування, що відділяє візуальне відображення та поведінку обробки подій у різні класи, а саме: представлення (View) та пред'явник (Presenter). А модель (Model) – клас для визначення даних, які будуть відображатися або над якими будуть проводитися інші дії у інтерфейсі користувача.

Приклади інтерфейсу розробленого мобільного застосунку наведено на рисунку 1.

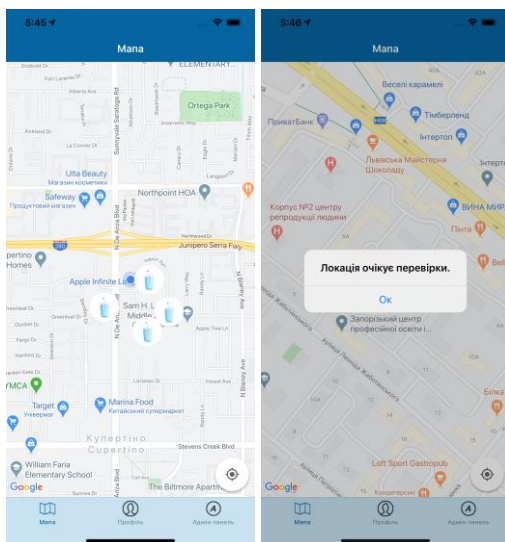


Рисунок 1 – Приклади інтерфейсу програми

В роботі був розроблений мобільний застосунок, що представляє собою карту пунктів безкоштовної видачі води з використанням мови програмування Swift в середовищі розробки XCode.

СЕКЦІЯ «СИСТЕМНИЙ АНАЛІЗ ТА ОБЧИСЛЮВАЛЬНА МАТЕМАТИКА»

УДК 311.21:519.25:614.2

Бахрушин В.Є.

д-р фіз.-матем. наук, проф. НУ «Запорізька політехніка»

ОСОБЛИВОСТІ СТАТИСТИЧНИХ ДАНИХ ДЛЯ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ЗАПОБІГАННЯ ПОШИРЕННЮ COVID-19

Основними показниками, якими оперують органи влади багатьох країн при прийнятті рішень щодо запобігання поширенню COVID-19, є кількості зареєстрованих, летальних і важких випадків, їх розподіл за групами, а також показники, що розраховують на їх основі. Втім, ці показники мають важливі особливості, які треба знати і враховувати.

Загальна кількість випадків, підтверджених тестуванням, завжди є меншою, ніж справжня кількість інфікованих. Політика багатьох країн, передбачає здійснення тестування лише у важких або симптоматичних випадках. Це залишає поза межами статистики невідому велику кількість осіб зі слабо вираженою або відсутньою симптоматикою. Якщо виходити з оцінок <http://epidemicforecasting.org>, то справжня кількість інфікованих в різних країнах, залежно від політики тестування, може бути в 3 – 30 разів вищою, за дані офіційної статистики. Для України цей показник станом на 02.04.2020 становив 24,2. Справедливість оцінок за моделями залежить від коректності припущень моделей, які наразі важко оцінити. Більш достовірні оцінки можна отримати, якщо робити тестування випадкових репрезентативних вибірок незалежно від наявності симптоматики. Втім, більшість країн вважає, що на сьогодні такі дослідження не є пріоритетними. Тому поки залишається лише констатувати, що справжня кількість інфікованих є невідомою, але значно більшою за офіційні показники. Це необхідно враховувати при побудові прогнозів розвитку пандемії і прийнятті рішень, що базуються на цих прогнозах.

Показники кількості летальних і важких випадків є більш достовірними, але і тут є нюанси. По-перше, більшість країн в офіційній статистиці наводять показник “Case related deaths”. Тобто не кількість летальних випадків, зумовлених COVID-19, а кількість летальних випадків в осіб, які мали позитивні результати тестів на COVID-19. Особливістю COVID-19 є те, що інфекція може істотно погіршувати стан здоров'я осіб, що мають інші захворювання, зокрема, хронічні захворювання серцево-судинної системи і легенів, онкологічні захворювання, діабет та ще деякі. За даними різних країн, частка померлих, що враховані у статистиці “Case related deaths” і мали

принаймні одне важке супутнє захворювання, становить від 80 до 98%. Тому у частині випадків, зареєстрованих як смертність від COVID-19, насправді COVID-19 є супутнім захворюванням, а смерть пацієнтів зумовлена іншими причинами. Надійних оцінок цієї частки на сьогодні немає по жодній країні. З іншого боку, є припущення, що через неповноту тестування не всі летальні випадки, зумовлені COVID-19, потрапляють до офіційної статистики. Аналогічною є ситуація з даними по важких випадках.

Показник смертності, який уряди багатьох країн використовують для прийняття рішень, отримують шляхом ділення кількості летальних випадків на кількість зареєстрованих хворих, або шляхом ділення кількості летальних випадків на суму кількостей летальних випадків і пацієнтів, що одужали. Це, відповідно, нижня та верхня оцінки показника “Case fatality rate” (CFR). Для країн, де кількість активних випадків становить незначний відсоток від загальної кількості зареєстрованих випадків, ці оцінки CFR є близькими одна до одної. Приміром, для Китаю, де кількість активних випадків становить 1,3% від загальної кількості, вони дорівнюють, відповідно, 4,06% та 4,13% (тут і далі оцінки розраховані за даними <https://www.worldometers.info/coronavirus> станом на 12.04.2020. З іншого боку, для США, де кількість активних випадків дорівнює 90,0%, оцінки CFR становлять, відповідно, 3,9% та 40,3%. Обидві оцінки є істотно різними для різних країн через відмінності політики тестування, спроможності медичних систем та різні стадії розвитку COVID-19. Зокрема, для Ісландії, де здійснюється одне з найбільш масових тестувань, вони становлять 0,47% та 0,94%, а для Італії, де діагностують лише важкі випадки – 12,8% та 37,4%. Тому оцінки CFR можна застосовувати для аналізу розвитку пандемії у конкретній країні за умови відсутності змін у політиці тестування, або для порівняння країн з однаковими політиками тестування. Але вони не придатні для прийняття рішень, що базуються на прогнозах справжньої захворюваності та кількості летальних і важких випадків.

Справжню частку летальних випадків характеризує показник “Infection fatality rate” (IFR), який є відношенням кількості летальних випадків до дійсної кількості інфікованих осіб. Через зазначені вище особливості вихідних даних, цей показник не може бути розрахований за даними офіційної статистики. На сьогодні використовують три способи його оцінювання – за моделями динаміки, за залежністю показника CFR від показників тестування та за вибірковими дослідженнями. У всіх випадках, останні оцінки свідчать, що IFR є значно меншим за первинні оцінки січня – лютого (кілька відсотків), і не перевищує кількох десятих відсотка і, можливо, є меншою за 0,1% [1, 2].

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Merelo J. J., Víctor Rivas-Santos [Електронний ресурс] / J. J. Merelo – Режим доступу: https://figshare.com/articles/Estimating_the_infection_fatality_rate_of_COVID_19_in_South_Korea_by_using_time_series_correlations/12083322
2. Michael Le Page [Електронний ресурс]– Режим доступу: <https://www.newscientist.com/article/2239497-why-we-still-dont-know-what-the-death-rate-is-for-covid-19>.

УДК 538.971

Корніч Г.В.¹, Широкоград Д.В.²

¹ д-р. фіз.-мат. наук, проф. НУ «Запорізька політехніка»

² канд. фіз.-мат. наук, ст. викл. НУ «Запорізька політехніка»

ЕВОЛЮЦІЯ ЯНУСОПОДІБНИХ Ni-AL КЛАСТЕРІВ ПІД ВПЛИВОМ НИЗЬКОЕНЕРГЕТИЧНИХ КЛАСТЕРІВ Ar та Ar₁₃

Класичним методом молекулярної динаміки моделюється еволюція вільних 390-атомних янусоподібних Ni-Al кластерів протягом 100 і 500 пс під впливом частинок Ar і Ar₁₃ з енергіями до 1,0 кЕВ. Початкові метастабільні янусоподібні кластери мають дві однокомпонентні частини, рівні за кількістю атомів, з невеликим просторовим перекриттям. Після удару снарядів починається швидке розширення кластерів-мішеней, що супроводжується розпиленням в умовах високотемпературного піку протягом перших 0,5 пс. При цьому, кінетики снарядів Ar₁ і Ar₁₃ критично відрізняються. Після цього кластери Ni-Al поступово розвиваються внаслідок екзотермічного змішування компонентів та демонструють тенденцію до поверхневої сегрегації атомів Al. Поверхнєве збагачення атомами Al після 500 пс чітко видно у випадку ударів Ar₁, тоді як при ударах Ar₁₃ енергіями понад 100 еВ ця тенденція маскується інтенсивним розпиленням, що включає як каскадно-рекойловий, так і тривалий тепловий внесок, з великим переважанням атомів Al у вихідному матеріалі.

Особливий інтерес становить дослідження еволюції структури біметалевих кластерів. Температури плавлення металевих нанокластерів зазвичай нижчі, ніж температури плавлення тих самих макроскопічних матеріалів, хоча вони можуть немонотонно змінюватись при зменшенні розміру кластера нижче двохсот атомів [1]. Значення AEI (індексів атомної еквівалентності), що є сумами модулів атомних радіус-векторних відстаней, були розраховані в нашому дослідженні за методикою Берендсена залежно від температури кластера і представлені на загальному рис. 1 для всіх атомів кластера.

Результати показують, що процес плавлення, який руйнує атомну структуру кластера, як очікувалося, починається з його поверхні за нижчих температур і сягає внутрішніх областей приблизно за 870 і 550 К для однокомпонентних кластерів Ni і Al, що складаються з 195 атомів кожен. Різниця в точках плавлення поверхневого та внутрішнього шарів в наших розрахунках сягають понад 200 К, особливо для кластеру Ni, як видно на рис. 1.

У наших моделюваннях для випадків бомбардування Ag_1 показали, що температури кластера Ni-Al після еволюції протягом 100 пс вищі за температури плавлення відповідних вільних однокомпонентних кластерів Al і Ni. Більше того, температури кластерів вищі, ніж температури плавлення кластерів Al та Ni вже після 5 та 15 пс при всіх енергіях бомбардування. Тому у багатьох комп'ютерних експериментах, алюмінієва частина янусоподібного кластера демонструє більш швидкий перехід у неупорядкований стан. У цих випадках протягом певного періоду часу частина нікелю частково покрита більш рухливими атомами алюмінію. Таким чином, протягом певного часу існує нестабільна для цієї атомної системи форма «ball-and-cup» компонентів [2]. Крім того, колізійна взаємодія кластера-мішені зі снарядом Ag_1 призводить до генерації радіаційних дефектів та руйнування впорядкованої структури атома в кластері за сильних нерівноважних умов протягом декількох початкових пікосекунд, що також сприяють утворенню розплаву. Таким чином, неструктурована форма, яку можна інтерпретувати як майже рідкий стан кластеру Ni-Al після взаємодії зі снарядом Ag_1 , передбачалася в цих моделюваннях. Зауважимо, що перехід до неструктурованої форми кластера також призводить до збільшення його потенційної енергії.

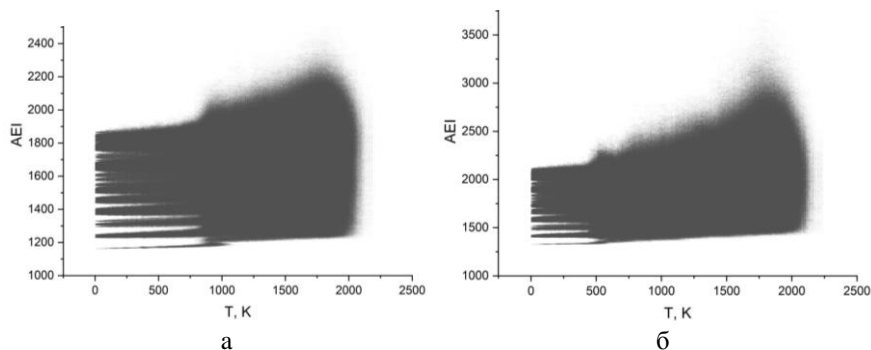


Рисунок 1 – Температурна залежність АЕІ для кластерів, що складаються з 195 атомів: а) Ni, б) Al.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Hamid I. Molecular dynamical simulations of melting behaviors of metal clusters / I. Hamid, M. Fang, H. Duan // AIP Adv. – 2015. – No. 5. – P. 047129.
2. Ferrando R. Symmetry breaking and morphological instabilities in core-shell metallic nanoparticles / R. Ferrando // J. Phys. Condens. Matter. – 2015. – No. 27. – P. 2713003.

УДК 519.85

Бакурова А.В.¹, Савранська А.В.², Шевчук М.В.³

¹ проф. НУ «Запорізька політехніка»

² доц. НУ «Запорізька політехніка»

³ магістр системного аналізу

ОПТИМІЗАЦІЯ ПОРТФЕЛЮ СЕЗОННИХ ТОВАРІВ

Дана робота присвячена математичному моделюванню оптимального портфелю сезонних товарів підприємства гуртової торгівлі. Проблема оптимізації є актуальною для підвищення життєздатності підприємства в умовах навали світової кризи. Як відомо, термінів "товарний асортимент" означає сукупність всіх асортиментних груп і товарних одиниць, що пропонуються для реалізації. Суміжне з ним поняття "товарний портфель" передбачає формування сукупності товарів підприємства за рівнем їх рентабельності, етапів життєвого циклу, сезонності, ринкових позицій та їхніх перспектив. Оптимальна структура товарного портфелю повинна забезпечувати водночас і фінансову стабільність підприємства, і вимоги споживачів та постачальників.

Моделюванню оптимальних портфелів присвячено велику кількість наукових публікацій, серед яких більшість спирається на класичну теорію Марковіца [1]. Розвитком класичної теорії виступають: сучасна теорія портфеля, темпоральна модель портфеля, теорія цифрового портфеля [2]. Одним з недоліків сучасної теорії портфелю, на відміну від цифрової, є те, що вона не заснована ні на якому часовому вимірі. Це призводить до того, що сучасна теорія портфеля породжує недоречні рішення, оскільки передбачає, що інвестори і фінансові ринки не мають часової залежності. В теорії цифрового портфеля, ризик і відносини кількісно визначаються на основі часових інтервалів. Теорія цифрового портфеля [2] представляє нову епоху аналізу і формування портфеля, що ґрунтується на кількісному аналізі ризику, прибутку та часу.

Прикладом використання цих теорій до побудови моделей оптимального портфелю є робота [3], де проведено дослідження комплексної

диверсифікації аптечної мережі в місті Запоріжжя, що передбачає мінімізацію ризиків у разі фактичного або потенційного зниження прибутковості аптек та інших чинників. В іншій роботі [4] показано, як при формуванні товарного портфеля на загальноукраїнському ринку використовують внутрішню організацію збуту підприємства. Зокрема, матриця у координатах «частка товару у збуті підприємства – участь товару у тренді збуту» дає змогу провести оцінку переваг та принадності ринку для кожного товару підприємства. Але зауважимо, що даний підхід не можна застосовувати до товарів з різним типом сезонності.

Метою даного дослідження є моделювання оптимального портфелю товарів з різним типом сезонності для зменшення залежності загальних обсягів збуту підприємства гуртової торгівлі від фактора сезонності.

Вихідні дані було взято на підприємстві гуртової торгівлі «Саванна», яка займається реалізацією трикотажних виробів. На рис.1 представлені обсяги продажів по групах: спідня білизна та колготки. Ці групи є найбільш вагомими в структурі підприємства.

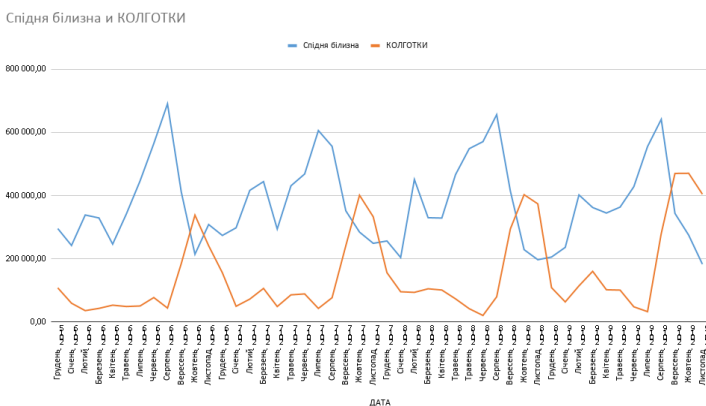


Рис. 1 – Динаміка обсягів збуту товарів з різним типом сезонності

На рис. 1 можна побачити, що максимуми та мінімуми продажів для груп товарів: спідня білизна та колготки не співпадають. Білизна найліпше продається влітку та погано продається восени. Продажі колготок навпаки різко зростають восени і майже не продаються влітку. Тому, можна зробити висновок, що продажі літа та осені є найбільш високими завдяки тому, що дві вищезазнані групи товарів компенсують одна одну у цей період. Найбільш низькі загальні продажі відбуваються у періоди, коли обидві групи (спідня білизна та колготки) погано продаються. Отже, для зниження чинника сезонності потрібно розширити асортимент додаванням груп товарів, які не

мають різко вираженої сезонності продажів (наприклад, рушники, постільна білизна), або товарів, максимумами і мінімумами продажів яких не співпадають з обсягами продажів основних груп товарів підприємства.

Для вирішення задачі запропоновано алгоритм оптимізації портфеля, що має такі етапи:

1. Вирішення задачі класифікації товарів за типом сезонності. В результаті виділено 4 типи. Позначимо частку i -го типу в портфелі:

x_1 - частка товарів, максимумами продажів яких припадають на весну і мінімумами – на осінь;

x_2 - частка товарів, максимумами продажів яких припадають на осінь і мінімумами – на весну;

x_3 - частка товарів, що не мають різко вираженої сезонності продажів;

x_4 - частка товарів, максимумами і мінімумами продажів яких не співпадають з обсягами продажів основних груп товарів.

2. Визначення факторів, що впливають на доходність товарів кожного типу, розрахунок факторних вагів, побудова рівнянь регресії.

3. Прогнозування динаміки обраних факторів.

4. Визначення очікуваної доходності та ступеня ризику від збуту кожного типу товару.

5. Визначення оптимального набору товарів та їх частки в портфелі при вирішенні двокритеріальної задачі оптимізації – максимізації доходності та мінімізації ризику.

Як інструмент управлінського впливу, модель оптимального товарного портфелю дозволяє сформувати оптимальну структуру переліку товарів, дає змогу розробити конкурентну стратегію підприємства, зменшити ризики та підвищити якість управлінських рішень. Перспективні напрями подальших досліджень визначаються врахуванням в моделі основних положень теорії цифрового портфеля та класифікації товарів за вектором ознак, що також включає сезонність, рентабельність, перспективність тощо в різних комбінаціях.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. H.M. Markowitz Risk–Return Analysis: The Theory and Practice of Rational Investing (a four-volume series) / H.M. Markowitz, K. Blay // McGraw-Hill, 2014.

2. Jones C.K., Modern Portfolio Theory, Digital Portfolio Theory and Intertemporal Portfolio Choice. / C.K. Jones // American Journal of Industrial and Business Management. 7,– 2017 – 833-854 p. – DOI:10.4236/ajibm. 2017.77059.

3. Бакурова А. В. Моделювання оптимального портфелю клієнтів централізованої аптечної мережі / Бакурова А. В., Ропало Г. М., Терещенко Е.

В. // – 2019 – DOI: doi:10.15587/2312-8372.2019.186789.

4. Поляк Ю. Я. Інноваційні підходи щодо формування товарного портфеля підприємства / Ю. Я. Поляк // – 2019. DOI: <https://doi.org/10.33990/2070-4011.59.2019.178439>.

УДК 519.87

Кривцун О. В.

канд. фіз.-мат. наук, доц. НУ «Запорізька політехніка»

ДОПУСТИМИ ГРАФИ ТА ЇХНІ НЕОБХІДНІ ТА ДОСТАТНІ ГІПЕРКУБИ

Нехай множини A і B знаходяться у *фрагментарному відношенні*, $A \prec_1 B$, якщо $A \subset B$ і $|B \setminus A| = 1$. У відповідності до [1] *фрагментарна структура* – це впорядкована пара (X, Φ) , де X – це скінченна множина, $\Phi = \{E_0, E_1, \dots, E_m\}$ – сімейство її підмножин, $E_i \subseteq X, i = 0, \dots, m$, таке, що $E_0 \equiv \emptyset$ і кожна підмножина з Φ знаходиться у фрагментарному відношенні хоча б з однією іншою підмножиною з Φ .

Оскільки фрагментарні структури є частково впорядкованими множинами, то їх можна представити як деякі підграфи орієнтованих гіперкубів, які будемо називати *допустимими* графами.

Гіперкуб мінімальної розмірності, підграфом якого є деякий допустимий граф, будемо називати *необхідним* гіперкубом [2] цього допустимого орграфа.

Будемо називати фрагментарні структури *ізоморфними*, якщо вони відповідають ізоморфним графам. Тоді кожний допустимий граф G відповідає класу F_G ізоморфних фрагментарних структур. Розв'язання задач підрахунку та переліку фрагментарних структур приводить до такої числової характеристики допустимого графа і відповідних йому ізоморфних фрагментарних структур, як розмірність фрагментарної структури.

Означення 1. *Розмірністю* N фрагментарної структури будемо називати величину

$$N = \min_{F_G} |\bigcup_{i=1}^m E_i|,$$

де $E_i, i = \overline{1, m}$ – фрагменти фрагментарної структури класу F_G .

Гіперкуб розмірності N будемо називати *достатнім* гіперкубом допустимого графа тієї ж розмірності [2].

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Kozin I. V., Perepelitsa Fragmentary Structures in Discrete Optimization Problems / I. V. Kozin, N. K. Maksyshko, V. A. // Cybernetics and Systems Analysis – 2017 – Vol. 53, Issue 6 – P. 931–936.

2. Кривцун О. В. Представлення фрагментарних структур орієнтованими графами. Кибернетика и системный анализ / О. В. Кривцун // междунар. науч.-теорет. журн. – 2019. – Т.55, № 2. – С. 1 – 9.

УДК 536.25

Денисенко О.І.¹

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

ЧИСЕЛЬНЕ МОДЕЛЮВАННЯ НЕСТАЦІОНАРНОГО ПРОЦЕСУ ВИМИВАННЯ ДОМІШОК З ВОДОЙМИЩ

Останнім часом спостерігається значний антропогенний вплив на навколишнє середовище. Виникають проблеми забруднення водних об'єктів промисловими стоками, що призводить до значного підвищення концентрації шкідливих речовин в воді. В результаті інтенсивного випаровування води в літній період може підвищуватись концентрація солей в замкнених водоймищах або в ставках охолоджувачах крупних енергетичних об'єктах, що призводить до негативних екологічних наслідків. Для зниження концентрації домішок ставки періодично промиваються чистою водою. В якості прикладу розглядається нестационарна задача вимивання домішок із водоймища прямокутної форми. Очищена вода подається через два отвори. На протилежній стороні водоймища знаходиться отвір, через який виходить забруднена вода. В якості математичної моделі використовувалась двовимірна система диференціальних рівнянь в наближенні Сен-Венана. Чисельна реалізація математичної моделі відбувалась за допомогою програмного пакету COMSOL Multiphysics. Для розрахунків використовувався метод скінчених елементів. За результатами розрахунків побудовані поля концентрацій забруднень та поля швидкостей. На рис.1 наведені приклади розрахунків концентрації домішок

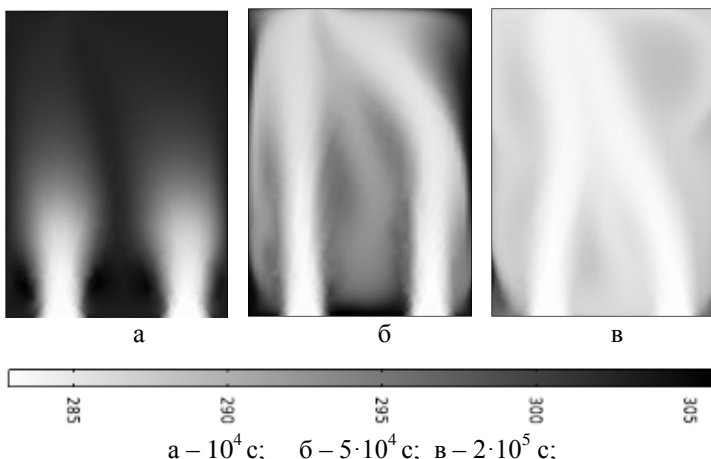


Рисунок 1 – Розподіл концентрації домішок для різних моментів часу з початку продукції.

УДК 512.56

Терещенко Е.В.¹, Литвиненко А.В.²

¹канд. фіз.-мат. наук, доц. НУ «Запорізька політехніка»

² студ. гр. КНТ-816 НУ «Запорізька політехніка»

ЗАДАЧА ПРО СПОРІДНЕНІСТЬ ЛІНІЙ СОНЯШНИКУ

Мета досліджень, що проводились у науковій сівозміні Інституту олійних культур НААН, це розробка методів визначення генетичної спорідненості ліній соняшнику за морфологічними ознаками. Дослідження 2016-2018 рр. містило 39 ліній соняшнику, які були оцінені за 34 морфологічними ознаками [1]. Для побудови графової деревовидної моделі було застосовано методи аналізу формальних понять (АФП) та теорії графів. На множині об'єктів G , які є лініями соняшників, множині M ознак ліній, з якими ведеться спостереження, та множині значень ознак W існує тернарне відношення $I \subseteq G \times M \times W$, що задається протоколом польового експерименту. На першому етапі необхідно побудувати решітку понять $B(G, M, W, I)$, що є множиною всіх понять контексту $K = (G, M, W, I)$, які упорядковані за вкладенням об'ємів. На другому етапі необхідно виокремити підграфи побудованої решітки, які є деревами та відображають спорідненість об'єктів. Для реалізації першого етапу розроблено алгоритм, який будує решітку так, щоб її елементи були розташовані по рівням, що робить його зрозумілим для

кінцевого користувача. У методі АФП для зображення решіток застосовуються лінійні діаграми, в яких ім'я об'єкта приписується найменшому з понять, в яких зустрічається цей об'єкт, а ім'я ознаки приписується найбільшому з понять, в яких присутня ця ознака. Вимоги до лінійної діаграми: прозорість, читабельність, легка інтерпретація даних. Реалізовано додаткову вимогу: симетричність зображення лінійних діаграм, побудованих по контексту та транспонованому контексту як ізоморфних. Для реалізації другого етапу, необхідно визначитися з критеріями спорідненості, які потребують подальших досліджень. Найпростіший критерій – це вважати, що родичі відрізняються найменшою кількістю ознак. Цікавим є введення відношення порядку на ознаках. Робота виконана в рамках договору про співробітництво між Інститутом олійних культур НААНУ та НУ«ЗІП», ДБ05028 «Аналіз біометричної інформації» (2018-2021pp).

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Махова Т.В. Використання ідентифікаційних морфологічних ознак у колекціях соняшнику / Махова Т.В., Ведмедєва К.В. // II інтернет конференції молодих учених «Генетика та селекція сільськогосподарських культур – від молекули до сорту»: тези конф., 30 серпня 2018р, м.Київ.– 2018. С.5-6.

УДК 519.213.1

Подковаліхіна О.О.¹, Січкара М.О.²

¹канд. фіз.-мат. наук, доц. НУ «Запорізька політехніка»

²студ. гр. КНТ-827сп НУ «Запорізька політехніка»

АНАЛІЗ РОЗВ'ЯЗКУ ЗАДАЧІ ЗАВАНТАЖЕННЯ З НОРМАЛЬНИМИ ПАРАМЕТРАМИ

Задача завантаження (інша назва – задача про рюкзак) є актуальною і досить затребуваною з точки зору її застосування в реальному житті. Це одна з задач максимізації, мета якої визначити як укласти якомога більше цінних речей в рюкзак, за умови, що загальний обсяг (або вага) всіх предметів, здатних поміститися в рюкзак, обмежений. Задача завантаження та її модифікації часто виникають в економіці, криптографії, лінгвістиці та логістиці для знаходження оптимального завантаження транспорту (літака, поїзда, трюму корабля) або складу [1,2]. Розглянемо наступну постановку задачі. Літак завантажуються предметами n різних типів. Кожен предмет типу i дає дохід c_i од. і важить a_i тон. Вантажопідйомність літака - b тон. Необхідно вибрати предмети, завантаження яких дозволить отримати максимальний дохід без перевищення вантажопідйомності літака [1]. Але на практиці доходи точно не відомі. Тому актуальним є розв'язання розглянутої

задачі в умовах невизначеності. Метою роботи є дослідження впливу статистичної невизначеності на оптимальний розв'язок задачі завантаження.

Як приклад розглянуто задачу з наступними вхідними даними: $n = 3$, $a_j = [3 \ 2 \ 1]$, $b = 5$, c_j підпорядковуються закону нормального розподілу з 50% відхиленням від значень $[45 \ 30 \ 15]$. Задачу було розв'язано методом перебору, який реалізовано за допомогою пакету MATLAB. Для наведених вхідних параметрів задачі було отримано чотири варіанти розв'язків, а саме (1,1,0), (0,0,5), (1,0,2), (0,2,1). Виникає питання, котрий план реалізовувати. Необхідно проаналізувати отримані розв'язки в залежності від кількості розрахунків (запусків програми) та різних критеріїв оптимальності. Наприклад, найбільший середньоарифметичний прибуток для 1000, 2000, 3000 та 4000 запусків дає план (0,0,5). Проте відзначимо, що оптимальні розв'язки, для різних критеріїв, можуть мінятися в залежності від кількості запусків, тому актуальним є питання, яка кількість експериментів є достатньою для визначення оптимального плану для заданого критерія або їх купності.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Таха, Хемді А. Т24 Введення в дослідження операцій / Хемді А. Таха – 7-е видання.: Пер. з англ. — М.: Видавничий дім "Вільямс", 2005. — 912 с.
2. Н.Ш. Кремер Фрідман: Дослідження операцій в економіці: Навч. посібник для вузів/ Н.Ш. Кремер, Б.А. Путко, И.М. Тришин М.Н. Фрідман – Під ред. проф. Н.Ш. Кремера. — М.: ЮНІТІ, 2005. - 407 с.

УДК 512.56

Білий В.В.

студ. гр. КНТ-827сп НУ «Запорізька політехніка»

СИСТЕМА УПРАВЛІННЯ ВЗАЄМОДІЄЮ З КЛІЄНТАМИ СТОМАТОЛОГІЇ

Надання стоматологічних послуг є тією областю, що пов'язана із застосуванням клієнтської бази і зберіганням значної кількості інформації. Виходячи з чого, виникла необхідність розробити спеціалізовану CRM-систему, що дозволить спростити виконання операцій, пов'язаних з обробкою даних і підвищити ефективність управління клінікою в цілому. До переваг такої системи можна віднести автоматизацію процесу запису пацієнтів на прийом та розрахунку вартості наданих послуг, можливість зберігання інформації стосовно історії відвідувань та перебігу лікування пацієнтів, складання індивідуальних комерційних пропозицій на основі аналітики, забезпечення взаємодії між співробітниками клініки, тощо.

З огляду на те, що даний програмний продукт призначений в першу чергу для корпоративного використання, було прийнято рішення реалізувати його у вигляді веб-додатку. Такий підхід дозволить персоналу вільно користуватися можливостями системи з будь-якого пристрою, що має доступ до мережі Інтернет.

Згідно останнім тенденціям веб-розробки, при написанні програми застосовувалася концепція SPA (односторінковий додаток). Особливість такого архітектурного рішення полягає в тому, що вся взаємодія з додатком відбувається на одній HTML-сторінці, а необхідні дані завантажуються динамічно за допомогою AJAX-запитів на сервер.

В порівнянні з іншими типами односторінкові додатки мають декілька переваг:

- при роботі завантажуються тільки необхідні дані (без повторюваних елементів і блоків), що дозволяє розвантажити ресурси сервера;
- відсутність повного перезавантаження сторінки підвищує швидкість роботи.
- код серверної частини (back-end) можна повторно використати для створення власної мобільної програми [1].

Ключове місце в цьому проекті займає база даних, на проектування якої слід звернути особливу увагу, оскільки існує потреба в роботі з великими обсягами інформації. Виходячи з цього під час розробки структури БД були використані механізми нормалізації. Нормалізація полягає в приведенні структури зберігання даних до нормальних форм. Нормальні форми визначають ряд вимог, що пред'являється до структури таблиць для забезпечення цілісності бази даних і усунення надмірних функціональних залежностей [2]. В цій системі база даних була приведена до перших трьох нормальних форм.

При розробці додатку з самого початку була визначена предметна область, враховувались побажання та рекомендації практикуючих стоматологів. Використання концепції SPA та хмарних технологій, у свою чергу, дозволяє створити єдину інформаційну, що дає можливість лікарям отримувати та вносити інформацію про пацієнтів безпосередньо на робочих місцях.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Single-page application vs multiple-page application. [Електронний ресурс] – Режим доступу: https://neoteric.eu/blog/single-page-application-vs-multiple-page-application/?utm_source=medium.com&utm_medium=social&utm_content=neo&utm_campaign=blog.

2. Нормалізація реляційних баз даних. [Електронний ресурс] – Режим доступу: <https://club.shelek.ru/viewart.php?id=177>

Наукове електронне видання
комбінованого використання
Можна використовувати в локальному та
мережному режимах

ТИЖДЕНЬ НАУКИ-2020.

Факультет комп'ютерних наук і технологій

Збірник тез доповідей щорічної
науково-практичної конференції серед студентів,
викладачів, науковців, молодих учених і аспірантів 13–17
квітня 2020 року

Один електронний оптичний диск (DVD-ROM);
супровідна документація.
Тираж 100 прим. Зам. № 540

Видавець і виготовлювач
Національний університет «Запорізька політехніка»
Україна, 69063, м. Запоріжжя, вул. Жуковського, 64 Тел.:
(061) 769–82–96, 220–12–14

Свідоцтво суб'єкта видавничої справи ДК № 6952 від 22.10.2019.