МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Кафедра ___системного аналізу та обчислювальної математики_____
(найменування кафедри)

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

_____Об'єктно-орієнтоване програмування _____
(назва навчальної дисципліни)

Освітня програма: _інтелектуальні технології та прийняття рішень в складних системах_____
(назва освітньої програми)

Спеціальність: _____124 Системний аналіз_____
(найменування спеціальності)

Галузь знань: _____12 Інформаційні технології_____
(найменування галузі знань)

Ступінь вищої освіти: _____бакалаврський_____
(назва ступеня вищої освіти)

Затверджено на засіданні кафедри
___СА та ОМ_____
(найменування кафедри)

Протокол №__7_ від __17.08.2020_____ р.

м. Запоріжжя _2020____

1. Загальна інформація	
Назва дисципліни	*Об'єктно-орієнтоване програмування*
Рівень вищої освіти	*Перший (бакалаврський) рівень*
Викладач	*Широкорад Дмитро Вікторович, к.ф.-м.н., ст. викладач*
Контактна інформація викладача	*Телефон кафедри 769-8247, Телефон викладача +380631182567, hoveringphoenix@gmail.com*
Час і місце проведення навчальної дисципліни	*Предметна аудиторія кафедри, комп'ютерні класи згідно розкладу занять*
Обсяг дисципліни	*розподіл годин: лекції-30, лабораторні заняття-30, вид контролю- залік,*
Консультації	*Згідно з графіком консультацій*

2. Пререквізити і постреквізити навчальної дисципліни

Пререквізити: основи математичного аналізу, лінійна алгебра, програмування та алгоритмічні мови

3. Характеристика навчальної дисципліни

Загальні компетентності:

- здатність до абстрактного мислення, аналізу та синтезу;
- здатність застосовувати знання у практичних ситуаціях;
- здатність до адаптації та дії в новій ситуації
- здатність вчитися і оволодівати сучасними знаннями;
- здатність до пошуку, оброблення та аналізу інформації з різних джерел.

Фахові компетентності:

- здатність формалізувати проблеми, описані природною мовою, у тому числі за допомогою математичних методів, застосовувати загальні підходи до математичного моделювання конкретних процесів;
- здатність використовувати системний аналіз як сучасну міждисциплінарну методологію, що базується на прикладних математичних методах та сучасних інформаційних технологіях і орієнтована на вирішення задач аналізу і синтезу технічних, економічних, соціальних, екологічних та інших складних систем.

Результати навчання:

В результаті вивчення дисципліни «*Об'єктно-орієнтоване програмування»* студент повинен

знати:

- основні ідеї та поняття сучасного програмування, об'єктно-орієнтованого підходу при розробці складних програм;
- сучасні методи аналізу, проектування та програмування складних програмних систем;
- об'єктно-орієнтовану мову програмування C++,
- характеристики інших мов програмування, їхні можливості та обмеження;
- принципи побудови новітніх візуальних середовищ для розробки програм із застосуванням візуального програмування;
- методи та засоби створення програм, що діють у інтерфейсному середовищі таких операційних систем, як Windows та UNIX.

вміти:

- виконати програмну реалізацію наданого алгоритму, проектування, розробку чи доробку програмного продукту проблемного або системного характеру із застосуванням ідей сучасного програмування, скласти програму мовою C/C++;
- створити інтерфейс програми із застосуванням засобів візуального програмування;
- створити бібліотечні засоби і проблемне розширення мови програмування;
- оволодіти у разі необхідності іншою мовою програмування, виконати вибір мови програмування для розв'язання проблеми.

4. Мета вивчення навчальної дисципліни

Метою викладання дисципліни "Об'єктно-орієнтоване програмування" є отримання студентом базових знань та практичних навичок з основ сучасної технології створення

складних програмних продуктів на базі ідей і принципів об'єктно-орієнтованого методу, візуального програмування. Такі знання можуть бути використані для побудови програм аналізу складних систем, а також у розробках як системного, так і прикладного програмного забезпечення ЕОМ з урахуванням сучасних вимог у відношенні до надійності, якості інтерфейсу та ефективності програмних продуктів. Отримані знання та практичні навички мають служити базою для опанування у подальшому нових майбутніх систем програмування

5. Завдання вивчення дисципліни

Завдання: *навчити студентів застосовувати сучасні мови програмування високого рівня, а також самостійно створювати власні прикладні програми з використанням концепцій об'єктно-орієнтованого програмування.*

6. Зміст навчальної дисципліни

У навчальній дисципліні «Об'єктно-орієнтоване програмування» розширюються знання студентів з технологій сучасного об'єктно-орієнтованого програмування за допомогою алгоритмічних мов на прикладі мови С++.Вивчається концепція об'єкту та класу, його компонентів та методів, атрибути доступу. Курс містить відомості про перевантаження операцій та обробку виключень.. Розглядаються поняття наслідування, поліморфізму та інкапсуляції. Вивчаються методи обробки виключень та базові підходи до візуального програмування.

7. План вивчення навчальної дисципліни

№ тижня	Назва теми	Форми організації навчання	Кількість годин
1.	**Сучасні технології створення програмного забезпечення ЕОМ** Елементи сучасної технології створювання програм і можливості сучасних мов та систем програмування. Етапи створення та життєвий цикл програмного продукту. Стандартизація мов програмування.	Лекції, лабораторні заняття, самостійна робота	4
2	**Базові поняття сучасного програмування** Алгоритми і способи їхнього подання. Поняття інформаційної одиниці, типу інформаційної одиниці, об'єкту. Атрибути об'єкту: адреса, значення, ідентифікатор, тип. Коротка характеристика і класифікація мов програмування. Рівні програмування. Мови низького та високого рівнів.	Лекції, лабораторні заняття, самостійна робота	4
3.	**Функції мови С/С++ та шаблони** Інтерфейс функції. Види параметрів функції: параметри-значення, параметри-покажчики, параметри-посилання. Прототипи функцій та їхнє застосування. Поліморфізм функцій, правило простого поліморфізму функцій. Шаблони функцій, їхнє застосування. Inline-функції, особливості їхнього виконання.	Лекції, лабораторні заняття, самостійна робота	4
4.	**Застосування синонімів і оператору typedef. Простори імен** Застосування оператору typedef для створення синонімів імен і функцій. Поняття простору імен, іменовані та не іменовані простори, що створюються автоматично.	Лекції, лабораторні заняття, самостійна робота	4

5.	**Класи, компоненти об'єкту, методи класу** Поняття класу, його загальна структура. Компоненти класу. Дані і функції класу. Статичні компоненти та функції класу. Класові типи та покажчики. Створення і робота функцій-компонентів класу (методів).	Лекції, лабораторні заняття, самостійна робота	4
6	**Конструктори, ініціалізація та перетворення об'єктів. Деструктори** Конструктори і деструктори. Поняття контейнерного та вкладеного класів. Перетворення типу для класів.	Лекції, лабораторні заняття, самостійна робота	4
7.	**Захист даних та атрибути доступу** Захист даних. Підвищення надійності програмування: атрибути доступу private, protected, public, їхня дія, переваги та недоліки їхнього застосування. Забезпечення доступу до компонентів класу із зовнішніх функцій, оголошення та застосування дружніх функцій.	Лекції, лабораторні заняття, самостійна робота	4
8.	**Перевантаження операцій. Перевантаження унарних та бінарних операцій** Перевантаження як форма поліморфізму. Перевантаження операції присвоєння. Динамічне присвоєння. Перевантаження операції індексації (доступу до елемента масиву) []. Перевантаження символів операцій "<<", ">>" для програмування операцій введення/виведення.	Лекції, лабораторні заняття, самостійна робота	4
9	**Успадкування та ієрархії класів. Віртуальні функції** Створення потоків ifstream, ofstream, зв'язування їх з файлами. Операції введення/виведення. Керування форматами. Безформатне введення/виведення. Файли з довільним доступом.	Лекції, лабораторні заняття, самостійна робота	4
10	**Абстрактні класи та чисті функції** Поліморфізм та абстрактні класи. Віртуальні функції у ієрархіях успадкування, їхнє призначення та побудова. Поняття чистої функції. Абстрактні класи та інтерфейс класу. Особливості абстрактних класів, їхня роль у побудові складних ієрархій класів.	Лекції, лабораторні заняття, самостійна робота	4
11	**Шаблони класів та їхня реалізація. Приклади побудови шаблонів класів** Шаблони класів та їхня реалізація. Побудова і використання шаблонів класів. Шаблони як абстрактні типи. Приклади побудови параметризованого класу. Клас для роботи з множинами. Спеціалізація класу з шаблонами. Спеціальне визначення шаблонного класу.	Лекції, лабораторні заняття, самостійна робота	4

12	**Стандартні бібліотеки програмних компонентів. Побудова бібліотечних модулів** Створення потоків ifstream, ofstream, зв'язування їх з файлами. Перевантаження потокових операцій, створення альтернативних засобів введення/виведення для класів. Переваги і недоліки стандартних потокових операцій введення/виведення. Побудова проблемно-орієнтованих бібліотечних модулів.	Лекції, лабораторні заняття, самостійна робота	4
13	**Обробка виключних ситуацій. Застосування конструкцій try-throw-catch** Застосування конструкцій try-throw-catch. Визначення блоку контролю за допомогою try. Генерування особливих ситуацій, оператор throw. Функції-обробники особливих ситуацій catch. Типи особливих ситуацій та засоби їх обробки. Побудова та застосування спеціальних класів для обробки виключних ситуацій. Множинна обробка виключень. Генерація виключень у функціях.	Лекції, лабораторні заняття, самостійна робота	4
14	**Системи візуального програмування** Створення програм для багатозадачної операційної системи. Програми, виконання яких керується подіями. Проблема створення віконного інтерфейсу прикладної програми. Системи візуального програмування Microsoft Visual Basic, Microsoft Visual C++, Borland C++ Builder. Принципи об'єктно-орієнтованого візуального програмування. Візуальне програмування інтерфейсу програми. Системи швидкої розробки програм. Структура головного файлу проекту. Структура файлів модулів форм. Доступ до об'єктів, змінних та функцій модулю.	Лекції, лабораторні заняття, самостійна робота	6

8. Самостійна робота

Кількість годин, що плануються на виконання завдань самостійної роботи, тем завдань, включно з темами курсових робіт, наведені у таблиці п. 4 та у п. 9.
Необхідним елементом успішного засвоєння навчального матеріалу дисципліни є самостійна робота студентів з вітчизняною та закордонною спеціальною літературою. Самостійна робота є основним засобом оволодіння навчальним матеріалом у час, вільний від обов'язкових навчальних занять.
Основні види самостійної роботи, запропоновані студентам:

- *вивчення лекційного матеріалу;*
- *робота з рекомендованою літературою;*
- *вивчення окремих тем або питань, що передбачені для самостійного опрацювання;*
- *вивчення основних термінів та понять з галузі регресійного аналізу;*
- *підготовка до лабораторних занять;*
- *підготовка до проміжного та підсумкового контролю;*
- *контрольна перевірка кожним студентом особистих знань за питаннями для самостійного поглибленого вивчення та самоконтролю;*
- *робота над індивідуальним завданням.*

9. Система та критерії оцінювання курсу

Проміжний контроль знань студентів включає такі заходи:
виконання та захист лабораторних робіт, що передбачені учбовим планом;
аудиторні контрольні роботи (2 за семестр), що проводяться перед кожним підсумковим контролем;
захист курсової роботи.
Підсумковий контроль знань студентів включає:
- залік у 4-му семестрі;
- захист курсової роботи (у 4 семестрі).
Оцінювання знань при проміжному контролю можливо таким чином:

Розподіл балів, які отримують студенти

За результатами роботи протягом семестру студент отримує:
за відмінний захист лабораторних робіт – 40 балів;
при відмінному виконані всіх завдань підсумкової контрольної роботи (контрольної роботи для заочної форми) – 30 балів;
при відмінному захисті курсової роботи – 30 балів.

10. Політика курсу

Академічна доброчесність. Студенти не видають за свої результати роботи інших людей. При використанні чужих ідей і тверджень у власних роботах обов'язково посилаються на використані джерела інформації. Під час оцінювання результатів навчання не користуються недозволеними засобами, самостійно виконують навчальні завдання, завдання поточного та підсумкового контролю.
Відвідування занять. Очікується, що всі студенти відвідають усі лекції та практичні заняття курсу. За згоди декана та викладача дозволяється перейти на індивідуальний графік занять. У будь-якому випадку студенти зобов'язані дотримуватися термінів виконання усіх видів робіт, передбачених робочою програмою курсу.