

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ФАХОВИЙ КОЛЕДЖ РАКЕТНО-КОСМІЧНОГО МАШИНОБУДУВАННЯ
ДНІПРОВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ імені ОЛЕСЯ ГОНЧАРА**

ЗАТВЕРДЖУЮ

Заступник директора з НР

_____ В.М. Любохинець

«___» _____ 20__ р.

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

ПРОГРАМА

**обов'язкової навчальної дисципліни
підготовки фахового молодшого бакалавра
спеціальності 121 «Інженерія програмного забезпечення»
освітньої програми «Розробка програмного забезпечення»**

Дніпро
2022 рік

РОЗРОБЛЕНО ТА ВНЕСЕНО: Фаховим коледжем ракетно-космічного машинобудування
Дніпровського національного університету імені Олеся Гончара

РОЗРОБНИКИ ПРОГРАМИ: Гапоненко Наталія Володимирівна, викладач спецдисциплін,
спеціаліст I категорії.

Обговорено та рекомендовано до використання у навчальному процесі на засіданні ЦК
програмної інженерії

« 1 » вересня 2022 року, протокол № 1

ВСТУП

Програма вивчення *обов'язкової* навчальної дисципліни «Об'єктно-орієнтоване програмування» складена відповідно до *освітньо-професійної програми фахової передвищої освіти*

спеціальності 121 «Інженерія програмного забезпечення»,
освітньо-професійного ступеня фахового молодшого бакалавра.

Предметом вивчення навчальної дисципліни є технологія об'єктно-орієнтованого програмування.

Міждисциплінарні зв'язки: дисципліни «Основи програмування та алгоритмічні мови», «Інструментальні засоби візуального програмування».

Програма навчальної дисципліни складається з таких змістових модулів:

- 1 Інкапсуляція та приховування інформації.
- 2 Перевантаження операторів, використання шаблонів.
- 3 Успадкування та повторне використання коду.
- 4 Агрегація та композиція.
- 5 Поліморфізм та віртуальні функції.

1 Мета та завдання навчальної дисципліни

1.1 Метою викладання навчальної дисципліни «Об'єктно-орієнтоване програмування» є ознайомлення студентів з поняттями об'єктно-орієнтованого програмування та вивчення об'єктно-орієнтованої методології проектування, надання можливості вивчити мову програмування C++, яка є базовою для об'єктно-орієнтованого програмування і призначена для роботи з класами та об'єктами а також вивчення основних принципів об'єктно-орієнтованого програмування – інкапсуляції, успадкування та поліморфізму. Отримані знання є підґрунтям для вивчення сучасних технологій візуального об'єктно-орієнтованого та подійно-орієнтованого програмування. В межах предмету студенти вивчають призначене для цього середовище розробки на прикладі Microsoft Visual C++ або іншого, знайомляться з його інструментарієм, а також зі способами побудови інтерфейсу та реалізації класів, організації проектів в середовищі розробки.

1.2 Основними завданнями вивчення дисципліни «Об'єктно-орієнтоване програмування» є навчити студентів принципам об'єктно-орієнтованого проектування, поняттям інкапсуляції та приховування інформації, принципам перевантаження методів і операторів, поняттям успадкування та поліморфізму. Студенти повинні навчитись розробляти інтерфейс та реалізацію класів, створювати програми-проекти для роботи з об'єктами в середовищі розробки на прикладі Microsoft Visual C++ або іншого, а саме: виходячи із поставленої задачі самостійно розробляти класи, організовувати роботу з об'єктами створених класів, створювати інтерфейс програми, визначати необхідні методи для роботи з об'єктами, створювати програмний код для реалізації методів класів та роботи програми. Студенти повинні використовувати основні принципи об'єктно-орієнтованого програмування, володіти прийомами створення, знищення об'єктів, перевантаження операцій, створення ієрархії об'єктів.

1.3 Відповідно до вимог освітньо-професійної програми студенти повинні

знати : принципи об'єктно-орієнтованого проектування та базові поняття об'єктно-орієнтованого програмування – клас, інтерфейс та реалізація класу, об'єкти, конструктори та деструктори, перевантаження операцій; принципи об'єктно-орієнтованого програмування – інкапсуляції та приховування інформації, принципи перевантаження методів і операторів, розподіл інтерфейсу та реалізації, поняття успадкування та поліморфізму, правила побудови ієрархії класів, правила побудови агрегації класів, конструкції мови програмування C++, призначені для роботи з класами та об'єктами а також типові правила їх застосування; принципи об'єктно-орієнтованого програмування в середовищі розробки на прикладі Microsoft Visual C++ або іншого; інструменти інтегрованого середовища розробки.

вміти : самостійно використати класи для опису об'єктів реального світу, завдаючи їх характеристики та поведінку, проектувати та реалізовувати структуру класів, що відповідають предметній області розв'язуваної задачі, створювати конструктори і деструктори класів, перевантажувати методи класу і оператори, використовуючи механізми успадкування реалізовувати системи класів, реалізовувати та застосовувати поліморфні методи в ланцюгах успадкування, створювати об'єкти на основі розроблених класів та використати їх у проектах в середовищі розробки на прикладі Microsoft Visual C++ або іншого; володіти основами методів та технологій об'єктно-орієнтованого програмування; застосовувати набуті знання для здійснення професійної діяльності при розробці, налагодженні та експлуатації програмного забезпечення; розробляти програми, які практично використовують створені класи та об'єкти.

На вивчення навчальної дисципліни відводиться 285 годин / 9,5 кредитів ECTS.

2 Інформаційний обсяг навчальної дисципліни

Змістовий модуль 1 *Інкапсуляція та приховування інформації*

Тема 1.1 Методологія об'єктно-орієнтованого програмування. Поняття класу

Методологія об'єктно-орієнтованого програмування. Підвищення рівня абстракції програми. Поняття класу. Структура та опис класу. Поняття класу. Специфікатори доступу

Тема 1.2 Поняття об'єкту. Інкапсуляція та приховування інформації

Поняття об'єкту. Конструктори та деструктори. Поняття об'єкту. Вказівник this. Функції-члени класу. Інкапсуляція та приховування інформації

Змістовий модуль 2 *Перевантаження операторів, використання шаблонів*

Тема 2.1 Дружні функції та класи. Перевантаження операторів

Перевантаження функцій. Дружні функції. Дружні класи. Використання дружніх функцій та класів для доступу до членів класу. Перевантаження операторів. Операторні функції-члени класу. Операторні функції-друзі класу

Тема 2.2 Статичні елементи класу. Шаблони

Статичні елементи класу. Шаблони функцій. Константні члени класу. Константні методи. Робота з масивами об'єктів. Перевантаження операцій введення-виведення

Змістовий модуль 3. Успадкування та повторне використання коду

Тема 3.1 Успадкування та повторне використання коду

Успадкування та повторне використання коду. Просте поодиноке успадкування. Побудова ієрархії класів. Конструктори та деструктори при успадкуванні. Перевизначення методів базового класу. Шаблони класів

Тема 3.2 Множинне успадкування

Множинне успадкування. Конструктори та деструктори при множинному успадкуванні. Конфлікт імен і протоколів класу. Контейнерні класи. Узагальнення знань і умінь студентів

Змістовий модуль 4 Агрегація та композиція

Тема 4.1 Агрегація та композиція класів

Агрегація та композиція класів. Особливості конструювання класів при агрегації. Відношення між класами. Діаграми класів

Тема 4.2 Розробка інтерфейсу додатку

Створення одновіконного візуального додатку. Робота з одновіконним візуальним додатком. Використання власних класів в візуальному додатку. Створення підключення до бази даних у візуальному додатку. Технології візуального об'єктно-орієнтованого та подійно-орієнтованого програмування

Змістовий модуль 5. Поліморфізм та віртуальні функції

Тема 5.1 Віртуальні функції. Механізм пізнього зв'язування

Віртуальні функції. Механізм пізнього зв'язування. Поліморфізм. Віртуальні деструктори. Абстрактні класи. Віртуальні класи. Поліморфні об'єкти.

Тема 5.2 Додатки на основі об'єктно-орієнтованого підходу

Розробка об'єктно-орієнтованого програмного забезпечення. Додатки на основі об'єктно-орієнтованого підходу. Розробка інтерфейсу додатку. Узагальнення знань і умінь студентів. Обов'язкова контрольна робота

Для придбання студентами практичних навичок розробки та реалізації програм з використанням об'єктно-орієнтованого підходу передбачені наступні лабораторні роботи:

Змістовий модуль 1 Інкапсуляція та приховування інформації

Тема 1.1 Методологія об'єктно-орієнтованого програмування. Поняття класу

Лабораторна робота №1. Знайомство з середовищем візуального програмування

Лабораторна робота №2. Створення простого проєкту

Лабораторна робота №3. Проєктування та створення класу

Лабораторна робота №4. Відлагодження та тестування програми, що містить

клас

Тема 1.2 Поняття об'єкту. Інкапсуляція та приховування інформації

Лабораторна робота №5. Створення програми з ініціалізацією полів в конструкторі класу

Лабораторна робота №6. Відлагодження програми з ініціалізацією полів в конструкторі класу

Лабораторна робота №7. Проектування функцій-членів класу

Лабораторна робота №8. Відлагодження та тестування програми з використанням функцій-членів класу

Лабораторна робота №9. Створення програми з використанням конструктора копіювання

Лабораторна робота №10. Відлагодження програми з використанням конструктора копіювання

Змістовий модуль 2 *Перевантаження операторів, використання шаблонів*

Тема 2.1 Дружні функції та класи. Перевантаження операторів

Лабораторна робота №11. Складання програми з перевантаженням імен функцій

Лабораторна робота №12. Відлагодження та тестування програми з перевантаженням імен функцій

Лабораторна робота №13. Проектування класу з дружніми функціями

Лабораторна робота №14. Відлагодження та тестування програми з дружніми функціями

Лабораторна робота №15. Складання програми з перевантаженням операторів методами класу та зовнішніми функціями

Лабораторна робота №16. Відлагодження та тестування програми з перевантаженням операторів

Тема 2.2 Статичні елементи класу. Шаблони

Лабораторна робота №17. Проектування класу з статичними членами-даними та статичними членами-функціями

Лабораторна робота №18. Відлагодження та тестування програми з статичними членами класу

Лабораторна робота №19. Складання програми з використанням шаблонів функцій

Лабораторна робота №20. Відлагодження програми з використанням шаблонів функцій

Змістовий модуль 3. *Успадкування та повторне використання коду*

Тема 3.1 Успадкування та повторне використання коду

Лабораторна робота №21. Складання програми з використанням простого успадкування класів

Лабораторна робота №22. Відлагодження та тестування програми з простим успадкуванням

Лабораторна робота №23. Проектування конструкторів класів при побудові ієрархії класів

Лабораторна робота №24. Відлагодження та тестування програми з перевизначенням методів базового класу

Лабораторна робота №25. Складання програми з використанням шаблону класу

Лабораторна робота №26. Відлагодження програми з шаблоном класу

Тема 3.2 Множинне успадкування

Лабораторна робота №27. Складання програми з множинним успадкуванням класів

Лабораторна робота №28. Відлагодження програми з множинним успадкуванням

Лабораторна робота №29. Складання програми з ромбовидною ієрархією класів

Лабораторна робота №30. Відлагодження програми з конфліктами при успадкуванні

Лабораторна робота №31. Складання програми з різними специфікаторами доступу при успадкуванні

Лабораторна робота №32. Відлагодження програми з різними специфікаторами доступу при успадкуванні

Змістовий модуль 4 *Агрегація та композиція*

Тема 4.1 Агрегація та композиція класів

Лабораторна робота №33. Складання програми з використанням композиції класів

Лабораторна робота №34. Відлагодження програми з агрегацією класів

Лабораторна робота №35. Створення програми з агрегацією або композицією класів

Лабораторна робота №36. Відлагодження програми з агрегацією або композицією класів

Тема 4.2 Розробка інтерфейсу додатку

Лабораторна робота №37. Створення одновіконного візуального додатку

Лабораторна робота №38. Відлагодження одновіконного візуального додатку

Лабораторна робота №39. Створення одновіконного візуального додатку з використанням власних класів

Лабораторна робота №40. Відлагодження одновіконного візуального додатку з використанням власних класів

Лабораторна робота №41. Створення програми з використанням масиву об'єктів

Лабораторна робота №42. Відлагодження програми з використанням масиву об'єктів

Лабораторна робота №43. Створення програми з підключенням до бази даних

Лабораторна робота №44. Відлагодження програми з підключенням до бази даних

Змістовий модуль 5. *Поліморфізм та віртуальні функції*

Тема 5.1 Віртуальні функції. Механізм пізнього зв'язування

Лабораторна робота №45. Створення програми з використанням віртуальних методів

Лабораторна робота №46. Відлагодження програми з використанням віртуальних методів

Лабораторна робота №47. Створення програми з використанням абстрактних класів

Лабораторна робота №48. Відлагодження програми з використанням абстрактних класів

Лабораторна робота №49. Створення візуального додатку з використанням статичних елементів класів

Лабораторна робота №50. Відлагодження візуального додатку з використанням статичних елементів класів

Тема 5.2 Додатки на основі об'єктно-орієнтованого підходу

Лабораторна робота №51. Відношення між класами в програмі. Створення програми

Лабораторна робота №52. Відношення між класами в програмі. Відлагодження програми

Лабораторна робота №53. Розробка інтерфейсу додатку. Створення програми

Лабораторна робота №54. Розробка інтерфейсу додатку. Відлагодження програми

Лабораторна робота №55. Створення багатовіконного візуального додатку

Лабораторна робота №56. Аналіз взаємозв'язку класів в багатовіконному візуальному додатку

Примірна тематика курсових проєктів

Програмний модуль для тестування з цивільного захисту

Програмний модуль "Фоторепортер" обліку замовлень фотосесій

Програмний модуль обліку проходження МСЕК

Програмний модуль ведення обліку продажу мобільних телефонів та аксесуарів

Програмний модуль ведення обліку роботи пункту обміну валюти

Програмний модуль обліку відправлення товарів гуманітарної допомоги

Програмний модуль "Бюджет міста" ведення обліку видатків місцевого бюджету

Програмний модуль "Реєстратура. Талон" замовлення талонів до лікарів

Програмний модуль ведення обліку розміщення переселенців

Автоматизація ведення бази даних обліку реєстрації новонароджених в РАГС

Програмний модуль каси самообслуговування магазину

Інтернет-магазин для замовлення піци "Pizza Van"

Автоматизація ведення бази даних проведення ультразвукових досліджень медичного закладу

Програмний модуль ведення обліку замовлень доставки їжі

Програмний модуль ведення обліку договорів на обслуговування багатоквартирних будинків

Автоматизація ведення бази даних лабораторних досліджень

Автоматизація ведення бази даних фізичних осіб-підприємців

3 Рекомендована література

1. Довбуш, Г. Ф. Visual C++ на примерах [Текст] / Г. Ф. Довбуш, А. Д. Хомоненко; под ред. А. Д. Хомоненко. – СПб. : БХВ-Петербург, 2007. -528 с.

2. Глушаков, С.В. Язык программирования C++. [Текст]:/ С.В. Глушаков, А.В.Коваль, С.В.Смирнов. - Х.:Фолио, 2001. - 500 с.

3. Лафоре, Р. Объектно-ориентированное программирование в C++ [Текст] / Р. Лафоре. – М. - СПб. : ПИТЕР, 2004.- 923 с.

4. Методичні рекомендації до виконання лабораторних робіт (затверджено, протокол засідання ЦК ПІ від 01.09.2022 р. №1).

5. Павловская, Т. А. C/C++. Программирование на языке высокого уровня [Текст] / Т. А. Павловская. – М. - СПб. : ПИТЕР, 2003. - 461 с.

7. Хортон, А. Visual C++ 2005: базовый курс. [Текст] / А.Хортон. – М. : ООО "И.Д. Вильямс", 2007. - 1152 с.

4 Форма підсумкового контролю успішності навчання – екзамен у III семестрі, екзамен у IV семестрі, виконання курсового проєкту у V семестрі.

5 Засоби діагностики успішності навчання – усні, письмові опитування; захист лабораторних робіт; обов’язкова контрольна робота.