

Міністерство освіти і науки України
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра комп'ютерних наук та інформаційних технологій (№ 302)

ЗАТВЕРДЖУЮ

Гарант освітньої програми


(підпис) Мирослав МОМОТ
(ініціали та прізвище)

«___» _____ 2024 р.

**СИЛАБУС ОBOB'ЯЗKОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

«Об'єктно-орієнтоване програмування»

(назва навчальної дисципліни)

Галузь знань: _____ 12 «Інформаційні технології»
(шифр і найменування галузі знань)

Спеціальність: _____ 122 «Комп'ютерні науки»
(код і найменування спеціальності)

Освітня програма: _____ «Комп'ютеризація обробки інформації та управління»
(найменування освітньої програми)

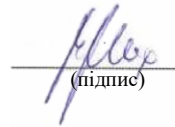
Форма навчання: денна

Рівень вищої освіти: перший (бакалаврський)

Вводиться в дію з «01» вересня 2024 р.

Харків 2024

Розробник: Мирослав МОМОТ, доцент, к.т.н., доцент
(прізвище та ініціали, посада, наукова ступінь та вчене звання)

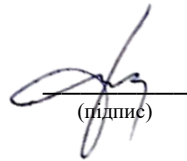

(підпис)

Силабус навчальної дисципліни розглянуто на засіданні кафедри (№ 302)
комп'ютерних наук та інформаційних технологій

(назва кафедри)

Протокол № 671/07 від « 27 » серпня 2024 р.

Завідувач кафедри Д.Т.Н., проф.
(науковий ступінь і вчене звання)


(підпис)

Олег ФЕДОРОВИЧ
(ім'я та прізвище)

Погоджено з представником здобувачів освіти:
студент групи 316ст


(підпис)

В'ячеслав БОГОНОС
(ім'я та прізвище)

1. Загальна інформація про викладача



Момот Мирослав Олександрович, к.т.н., доцент, доцент кафедри комп'ютерних наук та інформаційних технологій (№ 302). З 2001 з року викладає в університеті дисципліни, пов'язані з програмуванням, в даний час наступні:

- «Об'єктно-орієнтоване програмування»;
- «Сучасні технології програмування»;
- «Технологія створення програмних продуктів»;
- «Управління створенням програмних продуктів».

Напрями наукових досліджень: інформаційне моделювання при створенні й модернізації мереж зв'язку та передачі даних, інформаційне моделювання телекомунікаційних послуг, системи верифікації і моделювання технологічного процесу складання.

2. Опис навчальної дисципліни

Семестр, в якому викладається дисципліна – 3 семестр.

Обсяг дисципліни: 7,5 кредити ЄКТС (225 годин), у тому числі аудиторних – 80 годин, самостійної роботи здобувачів – 145 години.

Форми здобуття освіти - денна, дистанційна.

Дисципліна – обов'язкова.

Види навчальної діяльності – лекції, лабораторні заняття, самостійна робота здобувача.

Види контролю – поточний, модульний та підсумковий (семестровий) контроль (іспит).

Мова викладання – українська.

3. Мета та завдання навчальної дисципліни

Мета – дати знання про методи, принципи та концептуальні положення об'єктно-орієнтованого програмування для створення сучасних програмних продуктів.

Завдання – навчити студентів використовувати в практичній діяльності можливості об'єктно-орієнтованого програмування при створенні комп'ютерних систем

Компетентності, які набуваються:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК8. Здатність генерувати нові ідеї (креативність).

ЗК11. Здатність приймати обґрунтовані рішення.

ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

СК17. Здатність до розробки програмного забезпечення для задач управління об'єктами та процесами у реальному часі.

Очікувані результати навчання:

ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

ПР10. Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.

ПР11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).

ПР13. Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички

технології адміністрування комп'ютерних мереж та їх програмного забезпечення.

ПР14. Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.

ПР15. Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

ПР18. Виконувати розробку інструментальних засобів та програмного забезпечення для управління складними системами та процесами у реальному часі.

Пререквізити:

- ОК2. «Основи програмування»;
- ОК4. «Вступ до спеціальності»;
- ОК7. «Створення візуальних інтерфейсів»;
- ОК8. «Структури даних».

Кореквізити

- ОК13. «Мобільні та хмарні технології»;
- ОК14 «Веб-технології та веб-дизайн».

Постреквізити:

- ОК16. «Тестування програмних систем»;
- ОК18. «Мобільні та хмарні технології (КР)»;
- ОК19. «Ознайомча практика»;
- ОК21. «Операційні системи»;
- ОК24. «Статистичні та імовірнісні методи дата-аналізу (КР)»;
- ОК25. «Розробка веб-застосунків»;
- ОК28. «Технологія створення програмних продуктів»;
- ОК29. «Виробнича практика»;
- ОК31. «Створення систем штучного інтелекту та машинне навчання»;
- ОК32. «Проектування інформаційних систем»;
- ОК37. «Проектування інформаційних систем (КР)»;
- ОК38. «Кваліфікаційна робота».

4. Зміст навчальної дисципліни

Модуль 1.

Змістовний модуль 1.

Тема 1. Вступ до навчальної дисципліни «Об'єктно-орієнтоване програмування». Передумови появи об'єктно-орієнтованого програмування (ООП) для розробки складного програмного забезпечення

- *Форма занять: лекція, самостійна робота.*
- *Обсяг аудиторного навантаження: 6-10 годин.*
- *Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.*

Предмет, задачі та структура курсу. Зв'язок курсу з іншими дисциплінами спеціальності. Список рекомендованої літератури.

Складність, властива програмному забезпеченню. Організована та неорганізована складність. Канонічна форма складної програмної системи. Алгоритмічна та об'єктно-орієнтована декомпозиція. Роль декомпозиції, абстракції та ієрархії при розробці програмного забезпечення.

Покоління мов програмування. Топологія мов різного покоління. Структурна та процедурна парадигми програмування. Об'єктно-орієнтована парадигма програмування. Класифікація об'єктно-орієнтованих мов програмування.

- *Обсяг самостійної роботи здобувачів: 8-10 годин.*

Організована та неорганізована складність. Канонічна форма складної програмної системи. Структурна та процедурна парадигми програмування.

Опрацювання матеріалу лекцій. Формування питань до викладача. Вивчення термінів та визначень.

Тема 2. Структури та класи в C/C++/C#, Object Pascal/Delphi, Python.

- *Форма занять: лекція, лабораторна робота, самостійна робота.*
- *Обсяг аудиторного навантаження: 10-11 годин.*
- *Лабораторна робота: «Використання структур у C++/C# та записів в Object Pascal/Delphi».*

- *Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.*

Класи. Структури як класи. Характерні особливості, відмінності структур в C, C++, C#, а також записів в Object Pascal\Delphi. Виділення та визволення пам'яті. Особливості об'яви конструкторів та деструкторів та перевантаження конструкторів в C++, Object Pascal/Delphi, Python.. Функції-члени та дані-члени.

- *Обсяг самостійної роботи здобувачів: 14-16 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача. Вивчення термінів та визначень. Оформлення лабораторної роботи та підготовка до її здачі.

Тема 3. Інтерфейси та реалізації класів в C++/C#, Object Pascal/Delphi, Python. Области видимості.

- *Форма занять: лекція, лабораторна робота, самостійна робота.*
- *Обсяг аудиторного навантаження: 10-11 годин.*

- Лабораторна робота: «Використання класів в C++/C# та Delphi. Конструювання і знищення об'єктів. Динамічне виділення та визволення пам'яті».

- Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.

Інтерфейси та реалізації класів в C++. Області видимості. Показчик this. Приховання даних. Розділи класів. Друзі класу. Статичні члени. Вкладені класи.

- Обсяг самостійної роботи здобувачів: 14-16 годин.

Опрацювання матеріалу лекцій. Формування питань до викладача. Вивчення термінів та визначень. Оформлення лабораторної роботи та підготовка до її здачі.

Тема 4. Перевантаження операцій у мові C++/C#, Object Pascal/Delphi, Python.

- Форма занять: лекція, лабораторна робота, самостійна робота.

- Обсяг аудиторного навантаження: 10-11 годин.

- Лабораторна робота: «Перевантаження операцій у мові C++/C#».

- Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.

Бінарні й унарні операції. Наперед визначені значення операцій. Визначене перетворення типу. Обмеження на перевантаження операцій, особливості у мовах C++/C#, Object Pascal/Delphi, Python.

- Обсяг самостійної роботи здобувачів: 14-16 годин.

Опрацювання матеріалу лекцій. Формування питань до викладача. Вивчення термінів та визначень. Оформлення лабораторної роботи та підготовка до її здачі.

Модульний контроль 1

- Форма занять: написання модульної роботи в аудиторії (за рішенням лектора допускається проведення у дистанційній формі).

- Обсяг аудиторного навантаження: 2 години

- Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.

- Обсяг самостійної роботи здобувачів – за необхідністю (підготовка до модульного контролю).

Змістовний модуль 2.

Тема 5. Похідні класи. Побудова ієрархій класів та поліморфізм.

- Форма занять: лекція, лабораторна робота, самостійна робота.

- Обсяг аудиторного навантаження: 10-12 годин.

- Лабораторна робота: «Вивчення спадкування в ООП».

- Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.

Особливості об'яви похідних класів. Одиночне та множинне спадкування. Поля типу. Дерево та граф спадкування ієрархій класів. Віртуальні функції.

Поліморфна поведінка класів, які зв'язані спадкуванням. Чисті віртуальні функції. Абстрактний базовий клас.

- Обсяг самостійної роботи здобувачів: 10-12 годин.

Опрацювання матеріалу лекцій. Формування питань до викладача. Вивчення термінів та визначень. Оформлення лабораторної роботи та підготовка до її здачі.

Тема 6. Обробка виняткових ситуацій.

- Форма занять: лекція, лабораторна робота, самостійна робота.

- Обсяг аудиторного навантаження: 10-12 годин.

- Лабораторна робота: «Обробка виняткових ситуацій в C++/C#».

- Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.

Обробка помилок в стандартному C. Поняття винятку, виняткової ситуації стосовно програмного забезпечення. Оператори try, catch, throw. Вкладення блоків try-catch. Перехоплення виняткової ситуації. Оброблювач виняткової ситуації. Повторна генерація виняткової ситуації. Особливості обробки виняткових ситуацій у мовах C++/C#, Object Pascal/Delphi, Python.

- Обсяг самостійної роботи здобувачів: 10-12 годин.

Опрацювання матеріалу лекцій. Формування питань до викладача. Вивчення термінів та визначень. Оформлення лабораторної роботи та підготовка до її здачі.

Тема 7. Шаблони функцій та класів.

- Форма занять: лекція, лабораторна робота, самостійна робота.

- Обсяг аудиторного навантаження: 10-12 годин.

- Лабораторна робота: «Обробка виняткових ситуацій в C++/C#».

- Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.

Шаблони функцій, їх переваги та недоліки у порівнянні з макросами. Об'ява та виклик шаблонної функції. Шаблони класів. Об'ява та використання шаблонів класів. Параметри шаблонів. Інстацювання. Нетипові параметри шаблонів. Особливості узагальнень у мові C#.

Огляд шаблонів стандартної бібліотеки STL (Standard Template Library) в C++, а також узагальнених колекцій C# (System.Collections.Generic). Контейнерні типи, колекції, ітератори, черги, стеки.

- Обсяг самостійної роботи здобувачів: 10-12 годин.

Опрацювання матеріалу лекцій. Формування питань до викладача. Вивчення термінів та визначень. Оформлення лабораторної роботи та підготовка до її здачі.

Тема 8. Об'єктний підхід. Об'єктно-орієнтовані аналіз, проектування та програмування.

- Форма занять: лекція, лабораторна робота, самостійна робота.

- Обсяг аудиторного навантаження: 10-12 годин.

- Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.

Складові об'єктного підходу: абстрагування, інкапсуляція, модульність, ієрархія, слабка та сильна типізація, паралелізм, зберегаємість. Поняття об'єктно-орієнтованого аналізу (ООА), проектування (ООД) та програмування (ООР), їх взаємозв'язок.

Об'єкт як основа ООП. Стан та поведінка об'єкта, ідентичність об'єктів. Зв'язок між об'єктами – передача повідомлень. Ролі об'єктів: актор, сервер та агент. Класи. Відносини між класами. Різні розуміння поняття інтерфейсу, розділи інтерфейсу класу. Взаємозв'язок класів та об'єктів.

Асоціація як семантичний зв'язок між класами. Потужність зв'язку. Спадкування. Одиночне та множинне спадкування. Графічне позначення асоціації та спадкування. Приклади асоціації та спадкування. Використання. Клієнт та сервер при використанні. Агрегація. Фізичне включення як складова форма агрегації. Графічне позначення агрегації та використання. Приклади агрегації та використання. Інстанцування при використанні шаблонів. Графічне позначення інстанцування та метакласів. Приклади інстанцування та метакласів. Поняття класифікації як виділення класів та об'єктів предметної області. Проблеми класифікації.

Якість програмного продукту. Класичні міри якості програмного продукту. Залежність кількості знайдених помилок від часу. Щільність помилок. Якість класів та об'єктів. Об'єктно-орієнтовані міри якості програмного продукту. Зважена насиченість класу методами. Глибина дерева спадку. Число нащадків. Зачеплення класів. Відклик на клас. Нестача зв'язності у методах

Інструменти розробки програмного забезпечення. Набір інструментів попередніх поколінь мов програмування – редактор, компілятор, компоувальник, завантажувач. Набір інструментів об'єктно-орієнтованих мов програмування – графічний редактор, браузер класів, інкрементний компілятор, налагоджувач, компоувальник, завантажувач, інструменти управління конфігурацією та контролю версій проекту ПЗ, генератор графічного інтерфейсу користувача.

Перспективи розвитку об'єктно-орієнтованих аналізу, проектування та програмування.

- Обсяг самостійної роботи здобувачів: 8-10 годин.

Опрацювання матеріалу лекцій. Формування питань до викладача. Вивчення термінів та визначень.

Модульний контроль 2

- Форма занять: написання модульної роботи в аудиторії (за рішенням лектора допускається проведення у дистанційній формі).

- Обсяг аудиторного навантаження: 2 години

- Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.

- Обсяг самостійної роботи здобувачів – за необхідністю (підготовка до модульного контролю).

Модуль 2.

Індивідуальне завдання (РР).

- Форма занять: самостійна робота.
- Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.
- Обсяг самостійної роботи здобувачів: 40-45 годин.

5. Індивідуальні завдання

Виконання РР за варіантами на тему «Побудова діаграм взаємодії класів і об'єктів».

6. Методи навчання

Проведення аудиторних лекцій, лабораторних занять, індивідуальні консультації (при необхідності), самостійна робота студентів за матеріалами, опублікованими кафедрою (методичні посібники).

7. Методи контролю

Проведення поточного контролю, письмового модульного контролю, підсумковий контроль у вигляді іспиту.

8. Критерії оцінювання та розподіл балів, які отримують здобувачі

8.1. Розподіл балів, які отримують здобувачі (кількісні критерії оцінювання)

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Виконання і захист лабораторних робіт	0...10	3	0...30
Модульний контроль	0...15	1	0...15
Змістовний модуль 2			
Виконання і захист лабораторних робіт	0...10	3	0...30
Модульний контроль	0...15	1	0...15
Виконання і захист РР	0...10	1	0...10
Усього за семестр			0...100

Семестровий контроль (іспит) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту. Під час складання семестрового іспиту студент має можливість отримати максимум 100 балів.

Білет для іспиту складається з трьох запитань: двох теоретичних (максимальна кількість балів за повну та правильну відповідь на одне запитання - 30) та одного практичного (максимальна кількість балів - 40).

8.2. Якісні критерії оцінювання

Необхідний обсяг **знань** для одержання позитивної оцінки: основи об'єктно-орієнтованого аналізу, об'єктно-орієнтованого проектування; основні положення об'єктно-орієнтованого програмування, методи, способи та засоби розробки програм у рамках цього напрямку; сучасні методи та засоби розробки алгоритмів та програм, способи запису алгоритму мовою високого рівня, способи налагодження, випробування та документування програм; методи проектування та виробництва програмного продукту, принципи побудови, структури і прийоми роботи з інструментальними засобами, що підтримують створення програмного забезпечення.

Необхідний обсяг **вмінь** для одержання позитивної оцінки: формулювати проблему, ставити задачі та цілі; робити об'єктно-орієнтований аналіз предметної області; робити декомпозицію задачі у вигляді окремих класів та об'єктів, а також взаємозв'язків між ними; реалізовувати спроектовані класи у вигляді закінченого програмного забезпечення; працювати з інструментальними засобами, що підтримують створення програмного забезпечення в рамках об'єктно-орієнтованої парадигми програмування робити тестування, налагодження та документування програм.

8.3 Критерії оцінювання роботи здобувача протягом семестру

Задовільно (60-74). Показати мінімум знань та умінь. Відпрацювати та захистити всі лабораторні роботи та РР. Вміти застосовувати основи об'єктно-орієнтованого аналізу, знати основні положення об'єктно-орієнтованого програмування, методи, способи та засоби розробки програм у рамках цього напрямку. Вміти писати програмний код структур та класів.

Добре (75 - 89). Твердо знати мінімум знань, виконати усі завдання. Показати вміння виконувати та захищати всі лабораторні роботи в обумовлений викладачем строк з обґрунтуванням рішень та заходів, які запропоновано у роботах. Вміти робити об'єктно-орієнтований аналіз предметної області, робити декомпозицію задачі у вигляді окремих класів та об'єктів, а також взаємозв'язків між ними. Вміти пояснювати складні алгоритми, які реалізовані в програмах.

Відмінно (90 - 100). Повно знати основний та додатковий матеріал. Знати усі теми. Орієнтуватися у підручниках та посібниках. Досконально знати складові об'єктно-орієнтованого програмування, які використовуються при розробці складних програм. Вміти реалізовувати спроектовані ієрархії класів у вигляді закінченого програмного забезпечення; знати сучасні методи та засоби розробки алгоритмів та програм, способи запису алгоритму мовою високого рівня, способи налагодження, випробування та документування програм. Безпомилково виконувати та захищати всі лабораторні роботи в обумовлений викладачем строк з докладним обґрунтуванням рішень та заходів, які запропоновано у роботах.

Шкала оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційований залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

9. Політика навчального курсу

Відпрацювання пропущених занять відбувається відповідно до розкладу консультацій, за попереднім погодженням з викладачем. Питання, що стосуються академічної доброчесності, розглядає викладач або за процедурою, визначеною у Положенні про академічну доброчесність.

10. Методичне забезпечення

1. Навчально-методичне забезпечення дисципліни «Об'єктно-орієнтоване програмування». [Електронний ресурс] – Режим доступу: http://library.khai.edu/library/fulltexts/doc/Obyektno_Oriyentovane_Programuvanny_a.pdf

2. Електронні матеріали «Об'єктно-орієнтоване програмування» [Електронний ресурс] – Режим доступу: <https://mentor.khai.edu/course/view.php?id=1835>

11. Рекомендована література

Базова

1. Бублик В.В. Об'єктно-орієнтоване програмування: – К.: ІТкнига, 2015. – 624 с.

2. Alan Dennis, David Paul Tegarden, Barbara Haley Wixom. Systems Analysis and Design: An Object-Oriented Approach with UML– Wiley, 2015. – 544 p.

3. Andrew Troelsen. Pro C# 7: With .NET and .NET Core. - 8th ed. - Apress, 2017. - 1437 p.

4. Bjarne Stroustrup. Programming: Principles and Practice Using C++: 2nd edition. – Addison-Wesley Professional, 2014 –1312 p.

Допоміжна

1. І.М.Дудзяний. Об'єктно-орієнтоване моделювання програмних систем: Навчальний посібник. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2007. - 108 с.

2. Grady Booch. Object-Oriented Analysis and Design with Applications. – Addison-Wesley Professional, 2011. – 608 p.

12. Інформаційні ресурси

1. Об'єктно орієнтоване програмування (C#). [Електроний ресурс] – Режим доступу: <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/concepts/object-oriented-programming>