

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ

Декан факультету комп'ютерних
інформаційних технологій

 Ігор ЯКИМЕНКО

«» 2025 р.

ЗАТВЕРДЖУЮ

Проректор з науково-педагогічної
роботи

 Віктор ОСТРОВЕРХОВ

«» 2025 р.

ЗАТВЕРДЖУЮ:

Директор навчально-наукового
інституту новітніх освітніх технологій

 Святослав ПИТЕЛЬ

«» 2025 р.

РОБОЧА ПРОГРАМА

з дисципліни

«ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»

Ступінь вищої освіти – бакалавр

Галузь знань – 01 Освіта / Педагогіка

Спеціальність – 015 Професійна освіта (за спеціалізаціями)

Спеціалізація – 015.39 Цифрові технології

Освітньо-професійна програма – Професійна освіта (Цифрові технології)

Кафедра комп'ютерних наук

Форма навчання	Курс	Семестр	Лекції (год)	Практичні заняття (год)	ІРС (год)	Тренінг (год)	СРС (год)	Разом (год)	Екзамен (сем)
Денна	2	3	30	30	4	8	78	150	3
Заочна	2	3, 4	8	4	–	–	138	150	4

Тернопіль – ЗУНУ
2025

Робоча програма складена на основі освітньо-професійної програми підготовки бакалавра галузі знань 01 Освіта/Педагогіка, спеціальності 015 Професійна освіта (за спеціалізаціями), затвердженої Вченою радою ЗУНУ, протокол №11 від 26.06.2024 р. (зі змінами, протокол №8 від 26.06.2025 р.)

Робочу програму склав д.т.н, професор кафедри комп'ютерних наук Володимир МАНЖУЛА

Робоча програма затверджена на засіданні кафедри комп'ютерних наук, протокол № 1 від 26.08.2025 р.

Завідувач кафедри
д.т.н., проф.



Андрій ПУКАС

Гарант ОПП
к.е.н., доцент



Оксана БАШУЦЬКА

СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

"Об'єктно-орієнтоване програмування"

1. Опис дисципліни "Об'єктно-орієнтоване програмування"

Дисципліна – «Об'єктно-орієнтоване програмування»	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни
Кількість кредитів ECTS – 5	Галузь знань – 01 Освіта / Педагогіка	Статус дисципліни: обов'язкова Мова навчання: українська
Кількість залікових модулів – 5	Спеціальність – 015 Професійна освіта (за спеціалізаціями) Спеціалізація – 015.39 Цифрові технології	Рік підготовки: <i>Денна – 2</i> <i>Заочна – 2</i> Семестр: <i>Денна – 3</i> <i>Заочна – 3, 4</i>
Кількість змістових модулів – 2	Освітньо-професійна програма: Професійна освіта (Цифрові технології)	Лекції: <i>Денна – 30 год.</i> <i>Заочна – 8 год.</i> Практичні заняття: <i>Денна – 30 год.</i> <i>Заочна – 4 год.</i>
Загальна кількість годин: Денна: 150 год. Заочна: 150 год.	Ступінь вищої освіти – бакалавр	Самостійна робота: <i>Денна – 78 год.</i> <i>Заочна – 138 год.</i> Тренінг: <i>Денна – 8 год.</i> Індивідуальна робота: <i>Денна – 4 год.</i>
Тижневих годин: 10 год., з них аудиторних – 4 год.		Вид підсумкового контролю – екзамен

2. Мета й завдання вивчення дисципліни

"Об'єктно-орієнтоване програмування"

2.1. Мета вивчення дисципліни

Метою викладання дисципліни «Об'єктно-орієнтоване програмування» є формування у студентів професійних компетентностей, знань, умінь і навичок застосування об'єктно-орієнтованих мов програмування, зокрема в середовищі мобільної розробки (Kotlin для Android та Swift для iOS), для створення та використання цифрових освітніх продуктів у сфері професійної освіти.

2.2. Завдання вивчення дисципліни

Завдання вивчення дисципліни полягає в:

- одержанні студентами знань, умінь та навичок самостійної розробки об'єктно-орієнтованих програмних модулів, орієнтованих на вирішення задач професійної освіти;
- ознайомленні з принципами та засобами мобільної розробки на Kotlin (Android) та Swift (iOS) для створення освітніх додатків;
- формуванні вмінь працювати з інтегрованими середовищами розробки (Android Studio, Xcode) та налагоджувати програмний код;
- опануванні базових підходів до проєктування архітектури програмного забезпечення з використанням принципів ООП;
- розвитку здатності інтегрувати мобільні додатки в освітній процес професійно-технічних навчальних закладів.

2.3. В результаті вивчення дисципліни студенти повинні:

– **знати** основні парадигми програмування та сутність об'єктно-орієнтованого підходу; ключові принципи ООП: інкапсуляцію, спадкування, поліморфізм, абстракцію та їх роль у проєктуванні ПЗ; базові можливості та синтаксичні особливості сучасних ООП-мов (Java, C#, Python) і їх місце в екосистемі розробки; особливості мов Kotlin та Swift, структуру Android- та iOS-проєктів, основні компоненти мобільних додатків; принципи побудови інтерфейсу користувача та роботи з даними в мобільних освітніх додатках; можливості використання мобільних додатків у професійній освіті, вимоги до освітнього програмного забезпечення та базові аспекти безпеки й етики.

– **вміти** застосовувати принципи ООП під час проєктування та реалізації класів, ієрархій та програмних модулів; розробляти прості та проміжної складності мобільні додатки освітнього призначення на Kotlin (Android) та Swift (iOS); проєктувати та реалізовувати інтерфейси користувача, орієнтовані на потреби учнів і викладачів закладів професійної освіти; організовувати збереження та обробку даних у мобільних додатках із використанням відповідних засобів платформ (Room, SharedPreferences, Core Data, UserDefaults тощо); працювати з інтегрованими середовищами розробки, налагоджувати й тестувати створені додатки; планувати та реалізовувати навчальні сценарії з використанням власних мобільних освітніх додатків у практиці професійної освіти.

2.4. Найменування та опис компетентностей, формування котрих забезпечує вивчення дисципліни "Об'єктно-орієнтоване програмування"

K16. Здатність використовувати сучасні інформаційні технології та спеціалізоване програмне забезпечення та інтегрувати їх в освітнє середовище.

K19. Здатність використовувати відповідне програмне забезпечення для вирішення професійних завдань, відповідно до спеціалізації.

K25. Здатність збирати, аналізувати та інтерпретувати інформацію (дані) відповідно до спеціалізації.

2.5. Програмні результати навчання

ПР18. Розв'язувати типові спеціалізовані задачі, пов'язані з вибором матеріалів, виконанням необхідних розрахунків, конструюванням, проектуванням технічних об'єктів у предметній галузі (відповідно до спеціалізації).

ПР19. Уміти обирати і застосовувати необхідне устаткування, інструменти та методи для вирішення типових складних завдань у галузі (відповідно до спеціалізації).

3. Програма вивчення дисципліни

"Об'єктно-орієнтоване програмування"

Змістовий модуль 1 – Основи об'єктно-орієнтованого програмування

Тема 1. Вступ до об'єктно-орієнтованого програмування та його роль в професійній освіті

Парадигми програмування: історичний розвиток. Основні принципи ООП: інкапсуляція, спадкування, поліморфізм, абстракція. Порівняння процедурного та об'єктно-орієнтованого підходів. ООП-мови: класифікація та застосування (Java, C#, Python, Kotlin, Swift). Застосування ООП у професійній освіті: розробка навчальних додатків та тренажерів. Роль мобільних технологій у сучасних професійно-технічних закладах освіти.

Тема 2. Класи та об'єкти: фундаментальні концепції

Поняття класу як шаблону об'єктів. Властивості (атрибути) та методи класу. Створення об'єктів та їх життєвий цикл. Конструктори та деструктори. Модифікатори доступу (public, private, protected). Статичні члени класу та їх застосування в освітніх додатках.

Тема 3. Інкапсуляція та принципи приховування даних

Концепція інкапсуляції та її значення. Геттери і сеттери: правила використання. Властивості (Properties) у сучасних ООП-мовах. Інкапсуляція в контексті розробки безпечних освітніх додатків. Захист даних учнів та дотримання приватності. Практичні приклади інкапсуляції в мобільних додатках.

Тема 4. Спадкування та ієрархії класів

Концепція успадкування та повторного використання коду. Батьківські та дочірні класи. Перевизначення методів (method overriding). Ключове слово super/base. Множинне успадкування та його альтернативи. Проектування ієрархій класів для освітніх систем.

Тема 5. Поліморфізм та абстракція

Поліморфізм: compile-time та runtime. Перевантаження методів (method overloading). Віртуальні методи та динамічне зв'язування. Абстрактні класи та методи. Інтерфейси та їх роль у дизайні програм. Застосування поліморфізму в адаптивних освітніх додатках.

Тема 6. Огляд об'єктно-орієнтованих мов програмування

Java: платформа, екосистема, основний синтаксис ООП. С#: .NET Framework, особливості мови. Порівняльний аналіз Java та С#. Застосування Java та С# у розробці навчальних систем управління (LMS). Кросплатформенна розробка з використанням Xamarin. Приклади освітніх проектів на Java та С#. Особливості ООП у Python. Динамічна типізація та "качина типізація". Декоратори та метакласи. Python у сфері освіти: простота та доступність. Розробка прототипів освітніх додатків на Python. Інтеграція Python з мобільними платформами (Kivy, BeeWare).

Змістовий модуль 2 – Мобільна розробка та практичне застосування у професійній освіті

Тема 7. Kotlin як офіційна мова Android-розробки

Особливості синтаксису Kotlin. ООП у Kotlin: класи, data classes, sealed classes. Null-safety та безпека типів. Android Studio: середовище розробки. Структура Android-проекту та компоненти додатку.

Тема 8. Розробка інтерфейсу користувача на Android

Життєвий цикл Activity та Fragment. Layouts та View Groups (LinearLayout, ConstraintLayout). Material Design принципи для освітніх додатків. Робота з ресурсами (strings, drawables, styles). RecyclerView для відображення списків навчального контенту. Обробка подій та взаємодія з користувачем.

Тема 9. Робота з даними на Android: збереження та управління

SharedPreferences для налаштувань користувача. Room Database: локальне збереження даних учнів. SQLite та структуровані дані. Архітектурні паттерни: MVVM (Model-View-ViewModel). Робота з API та мережевими запитами (Retrofit). Синхронізація даних у навчальних додатках.

Тема 10. Swift як сучасна мова Apple

Swift як сучасна мова Apple. Синтаксис Swift та його відмінності від Kotlin. ООП у Swift: classes, structs, enums. Протоколи (Protocols) та розширення (Extensions). Xcode: середовище розробки для iOS. Структура iOS-проекту та компоненти.

Тема 11. Розробка інтерфейсу користувача на iOS

UIKit vs SwiftUI: сучасні підходи. Storyboards та програмне створення UI. View Controllers та навігація. Auto Layout для адаптивного дизайну. TableView та collectionView для навчального контенту. Human Interface Guidelines для освітніх додатків.

Тема 12. Робота з даними на iOS: збереження та управління

UserDefaults для простих налаштувань. Core Data для складних структур даних. CloudKit для синхронізації між пристроями. URLSession для мережових запитів. Архітектурні паттерни на iOS: MVC, MVVM. Організація даних у навчальних iOS-додатках.

Тема 13. Публікація та підтримка мобільних освітніх додатків

Google Play Console: публікація Android-додатків. App Store Connect: публікація iOS-додатків. Вимоги до освітніх додатків у магазинах. Монетизація та безкоштовні освітні рішення. Аналітика використання додатків (Firebase,

App Analytics). Підтримка та оновлення освітніх мобільних додатків. Етичні аспекти розробки освітнього софту.

4. Структура залікового кредиту дисципліни

"Об'єктно-орієнтоване програмування"

(денна форма навчання)

Теми	Кількість годин					
	Лекції	Практичні заняття	ІРС	Тренінг	СРС	Контрольні заходи
Змістовий модуль 1 – Основи об’єктно-орієнтованого програмування						
Тема 1. Вступ до об’єктно-орієнтованого програмування та його роль в професійній освіті	2	2	2	4	6	Поточне опитування
Тема 2. Класи та об’єкти: фундаментальні концепції	2	2			6	Поточне опитування
Тема 3. Інкапсуляція та принципи приховування даних	2	2			6	Поточне опитування
Тема 4. Спадкування та ієрархії класів	2	1			6	Поточне опитування
Тема 5. Поліморфізм та абстракція	2	2			6	Поточне опитування
Тема 6. Огляд об’єктно-орієнтованих мов програмування	2	1			4	Поточне опитування
Змістовий модуль 2 – Мобільна розробка та практичне застосування у професійній освіті						
Тема 7. Kotlin як офіційна мова Android-розробки	4	2	2	4	8	Поточне опитування
Тема 8. Розробка інтерфейсу користувача на Android	2	4			6	Поточне опитування
Тема 9. Робота з даними на Android: збереження та управління	2	4			6	Поточне опитування
Тема 10. Swift як сучасна мова Apple	4	2			8	Поточне опитування
Тема 11. Розробка інтерфейсу користувача на iOS	2	4			6	Поточне опитування
Тема 12. Робота з даними на iOS: збереження та управління	2	4			6	Поточне опитування
Тема 13. Публікація та підтримка мобільних освітніх додатків	2	-			4	Поточне опитування
Всього	30	30	4	8	78	150

(заочна форма навчання)

Теми	Кількість годин			
	Лекції	Практичні заняття	СРС	Контрольні заходи
Змістовий модуль 1 – Основи об’єктно орієнтованого програмування				
Тема 1. Вступ до об’єктно-орієнтованого програмування та його роль в професійній освіті	4	2	10	поточне опитування
Тема 2. Класи та об’єкти: фундаментальні концепції			12	поточне опитування
Тема 3. Інкапсуляція та принципи приховування даних			10	поточне опитування
Тема 4. Спадкування та ієрархії класів			12	поточне опитування
Тема 5. Поліморфізм та абстракція			12	поточне опитування
Тема 6. Огляд об’єктно-орієнтованих мов програмування			10	поточне опитування
Змістовий модуль 2 – Мобільна розробка та практичне застосування у професійній освіті				
Тема 7. Kotlin як офіційна мова Android-розробки	4	2	12	поточне опитування
Тема 8. Розробка інтерфейсу користувача на Android			12	поточне опитування
Тема 9. Робота з даними на Android: збереження та управління			10	поточне опитування
Тема 10. Swift як сучасна мова Apple			10	поточне опитування
Тема 11. Розробка інтерфейсу користувача на iOS			10	поточне опитування
Тема 12. Робота з даними на iOS: збереження та управління			10	поточне опитування
Тема 13. Публікація та підтримка мобільних освітніх додатків			8	поточне опитування
Всього	8	4	138	150

5. Тематика практичних занять

Практичне заняття 1-2

Вступ до ООП: перші кроки з класами та об'єктами

Мета: сформувати базові навички роботи з класами та об'єктами, застосовуючи принципи ООП для моделювання сутностей професійної освіти.

- Налаштування середовищ розробки (IntelliJ IDEA/PyCharm для загального ознайомлення).
- Створення простих класів на Java, C#, Python (клас "Учень", "Навчальний курс").
- Створення об'єктів та виклик методів.
- Реалізація простого консольного додатку для управління списком студентів.
- Знайомство з UML-діаграмами класів для моделювання освітніх систем.
- Обговорення застосування ООП у розробці навчального контенту.

Практичне заняття 3-5

Інкапсуляція, спадкування та поліморфізм у дії

Мета: закріпити вміння застосовувати інкапсуляцію, спадкування та поліморфізм під час проєктування і реалізації ієрархій класів для освітніх систем.

- Розробка системи класів "Навчальний Матеріал" з підкласами "Відео", "Текст", "Тест", "Інтерактивне завдання".
- Реалізація інкапсуляції: приватні поля, публічні методи доступу.
- Створення ієрархії класів з використанням спадкування.
- Перевизначення методів для різних типів навчальних матеріалів.
- Демонстрація поліморфізму: колекція різних навчальних матеріалів з єдиним інтерфейсом.
- Розробка абстрактного класу "Оцінювання" та його конкретних реалізацій для закладів професійної освіти.
- Практичне застосування: система обліку успішності студентів.

Практичне заняття 6

Основи Android-розробки на Kotlin

Мета: сформувати початкові навички розробки Android-додатків мовою Kotlin, зосереджуючись на створенні простих освітніх застосунків.

- Встановлення та налаштування Android Studio.
- Створення першого Android-проєкту з Empty Activity.
- Вивчення структури проєкту: Manifest, Gradle файли, ресурси.
- Написання простого класу на Kotlin з використанням data class.
- Створення інтерфейсу з TextView, EditText, Button.
- Реалізація обробки кліків та відображення введених даних.
- Розробка простого додатку "Калькулятор оцінок" для викладачів закладів професійної освіти.
- Тестування додатку на емуляторі або фізичному пристрої.

Практичне заняття 7-8

Розробка інтерфейсу навчального Android-додатку

Мета: розвинути вміння проєктувати та реалізовувати інтерфейс користувача Android-додатків, орієнтованих на відображення навчального контенту.

- Проєктування багатоекранного додатку з Navigation Component.
- Створення макету екрану зі списком навчальних тем (RecyclerView).
- Реалізація детального екрану окремої теми з описом та матеріалами.
- Застосування Material Design компонентів: CardView, FloatingActionButton.
- Робота з різними Layout: ConstraintLayout для складних екранів.
- Адаптація інтерфейсу під різні розміри екранів (responsive design).
- Розробка прототипу додатку "Мій навчальний журнал".
- Імплементация темної теми (Dark Mode) для комфортного читання.

Практичне заняття 9-10

Робота з локальною базою даних на Android

Мета: сформувати навички проєктування та реалізації локальної бази даних в Android-додатках для збереження та обробки освітньої інформації.

- Проєктування схеми бази даних для збереження інформації про учнів, курси, оцінки.
- Налаштування Room Database: створення Entity, DAO, Database класів.
- Реалізація CRUD операцій (Create, Read, Update, Delete).
- Використання ViewModel та LiveData для реактивного оновлення UI.
- Збереження налаштувань додатку через SharedPreferences.
- Робота з AsyncTask/Coroutines для асинхронних операцій з БД.
- Створення функціоналу "Облік відвідуваності".
- Експорт даних у CSV-формат для звітності.

Практичне заняття 11

Основи iOS-розробки на Swift

Мета: набути базових умінь розробки iOS-додатків мовою Swift для підтримки навчальної діяльності в закладах професійної освіти.

- Встановлення Xcode на macOS (або робота через хмарні сервіси/віртуалізацію).
- Створення першого iOS-проєкту.
- Вивчення структури iOS-проєкту: ViewController, Storyboard, Assets.
- Написання класів на Swift з використанням structs та classes.
- Робота з Optionals та безпечне розгортання значень.
- Створення простого інтерфейсу з UILabel, UITextField, UIButton.
- Розробка додатку "Планувальник занять".
- Симуляція роботи на різних iOS-пристроях (iPhone, iPad).

Практичне заняття 12-13

Розробка інтерфейсу навчального iOS-додатку

Мета: розвинути навички створення зручного та адаптивного інтерфейсу користувача в iOS-додатках освітнього призначення.

- Робота зі Storyboards: створення сцен та переходів (segues).
- Реалізація UINavigationController для навігації між екранами.
- Створення списку навчальних матеріалів за допомогою UITableView.
- Детальний перегляд матеріалу з UIScrollView для великого тексту.
- Застосування Auto Layout constraints для адаптивного дизайну.
- Робота з UITabBarController для організації розділів додатку.
- Розробка прототипу "Бібліотека навчальних ресурсів" для закладу професійної освіти.
- Використання SF Symbols для іконок.

Практичне заняття 14-15 **Робота з даними та мережею на iOS**

Мета: сформувати вміння працювати з даними та мережевими сервісами в iOS-додатках для організації доступу до навчальних ресурсів та збереження результатів навчання.

- Налаштування Core Data: створення Data Model, NSManagedObject.
- Збереження та завантаження даних з Core Data.
- Використання UserDefaults для зберігання налаштувань.
- Виконання мережових запитів через URLSession до освітніх API.
- Парсинг JSON-даних та відображення у додатку.
- Реалізація офлайн-режиму з кешуванням даних.
- Створення функції "Завантаження тестових завдань з сервера".
- Інтеграція з Firebase для синхронізації прогресу навчання між пристроями.

6. Самостійна робота

Самостійна робота як основна форма засвоєння студентом навчального матеріалу в час, вільний від обов'язкових занять та без участі викладача, передбачає особистісно-орієнтовану організацію самоосвіти студента. На самостійне опрацювання виносяться ряд тем, які студент повинен проаналізувати і на основі обраної написати доповідь (реферат) та підготувати презентацію.

Студенти мають можливість самостійно опрацьовувати навчальний матеріал у бібліотеці, навчальних кабінетах і комп'ютерних класах з використанням Інтернет, а також у домашніх умовах (при отриманні відповідного пакету навчально-методичної літератури).

№	Назва теми для самостійного опрацювання
1	Композиція vs спадкування: коли використовувати кожен підхід у проектуванні класів
2	Патерни проектування (Design Patterns): Singleton, Factory, Observer у мобільній розробці
3	Обробка винятків (Exception Handling) у Kotlin та Swift для надійних додатків

4	Generics (узагальнені типи) та їх застосування для типобезпечного коду
5	Lambda-вирази та функції вищого порядку у Kotlin та Swift
6	Coroutines у Kotlin: асинхронне програмування для Android
7	Async/Await у Swift: робота з асинхронним кодом на iOS
8	Jetpack Compose: сучасний декларативний UI для Android
9	SwiftUI: декларативне програмування інтерфейсів для iOS
10	Firebase Authentication: реалізація реєстрації та входу користувачів
11	Push-нотифікації для освітніх нагадувань на Android та iOS
12	Localization (локалізація): створення багатомовних освітніх додатків
13	Accessibility (доступність): проектування додатків для учнів з особливими потребами
14	Unit-тестування у мобільній розробці: JUnit, XCTest
15	Continuous Integration/Continuous Deployment (CI/CD) для мобільних проектів
16	GraphQL vs REST API: сучасні підходи до роботи з серверними даними
17	Кросплатформенна розробка: Flutter, React Native як альтернатива нативній розробці
18	Безпека мобільних додатків: шифрування даних, безпечне зберігання паролів
19	Гейміфікація у навчальних додатках: застосування ігрових елементів для мотивації
20	Аналіз user experience (UX): методи дослідження зручності освітніх додатків

7. Організація і проведення тренінгу

Тренінг з дисципліни "Об'єктно-орієнтоване програмування" полягає у розробці мобільного освітнього додатку для закладів професійної освіти. Мета: інтегрувати набуті знання та навички об'єктно-орієнтованого програмування, мобільної розробки на Kotlin (Android) або Swift (iOS) для створення повноцінного цифрового продукту, призначеного для підтримки навчального процесу в закладах професійної освіти.

Методика проведення тренінгу

Етап 1. Вибір теми та планування. Необхідно обрати тип освітнього додатку; визначити цільову аудиторію; обрати платформу розробки (Android з Kotlin або iOS зі Swift); скласти технічне завдання з описом функціональності та сформулювати очікувані результати використання додатку в освітньому процесі.

Етап 2. Проектування архітектури та дизайну. Необхідно створити UML-діаграму класів із визначенням основних сутностей та зв'язків між ними; спроектувати ієрархію класів із застосуванням принципів ООП (інкапсуляція,

спадкування, поліморфізм); розробити wireframes (ескізи) екранів додатку; спроектувати структуру бази даних (таблиці, поля, зв'язки).

Етап 3. Розробка моделі даних та бізнес-логіки. Необхідно реалізувати класи моделі даних із дотриманням принципів інкапсуляції; створити ієрархії класів з використанням спадкування та поліморфізму для різних типів навчальних матеріалів; налаштувати локальну базу даних; реалізувати CRUD-операції (створення, читання, оновлення, видалення) та протестувати бізнес-логіку окремо від інтерфейсу.

Етап 4. Розробка інтерфейсу користувача. Необхідно створити головний екран із навігацією між розділами додатку; реалізувати екрани для відображення списків навчального контенту (RecyclerView для Android або UITableView для iOS); розробити форми введення даних із валідацією; застосувати принципи Material Design (Android) або Human Interface Guidelines (iOS); забезпечити адаптивність інтерфейсу під різні розміри екранів.

Етап 5. Інтеграція, тестування та оформлення проєкту. Необхідно з'єднати інтерфейс із моделлю даних та налаштувати збереження і завантаження інформації; виконати функціональне тестування всіх можливостей додатку на різних пристроях або емуляторах; виправити виявлені помилки та обробити граничні випадки; оформити проєктну документацію (опис функціональності, діаграма класів, інструкція користувача, обґрунтування застосування додатку в професійній освіті); підготувати презентацію та провести демонстрацію роботи додатку.

8. Методи навчання

У навчальному процесі застосовуються: лекції, практичні та індивідуальні заняття, консультації, самостійна робота, робота у групах, методи опитування, моделювання практичних ситуацій, реферування, виконання індивідуальних завдань, підготовка і презентація проєктів.

9. Засоби оцінювання та методи демонстрування результатів навчання

У процесі вивчення дисципліни "Об'єктно-орієнтоване програмування" використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- поточне опитування;
- модульне оцінювання;
- презентації результатів виконаних завдань на тренінгу;
- оцінювання самостійної роботи;
- письмова самостійна робота;
- екзамен.

10. Політика щодо оцінювання

Політика щодо дедлайнів і перескладання. Для виконання усіх видів завдань студентами і проведення контрольних заходів встановлюються

конкретні терміни. Перескладання модулів відбувається з дозволу дирекції факультету за наявності поважних причин (наприклад, лікарняний).

Політика щодо академічної доброчесності. Списування під час проведення контрольних за ходів заборонені. Під час контрольного заходу студент може користуватися лише дозволеними допоміжними матеріалами або засобами, йому забороняється в будь-якій формі обмінюватися інформацією з іншими студентами, використовувати, розповсюджувати, збирати варіанти контрольних завдань.

Політика щодо відвідування. За об'єктивних причин (наприклад, карантин, воєнний стан, хвороба, закордонне стажування тощо) навчання може відбуватись в дистанційній формі за погодженням із керівником курсу з дозволу дирекції факультету.

11. Політика щодо визнання результатів навчання

Відповідно до «Положення про визнання в Західноукраїнському національному університеті результатів попереднього навчання»

(https://www.wunu.edu.ua/pdf/pologenya/Polozhennya_ruzult_poper_navch.pdf)

здобувачам вищої освіти може бути зараховано результати навчання (неформальної/інформальної освіти, академічної мобільності тощо) на підставі підтвердних документів (сертифікати, довідки, документи про підвищення кваліфікації тощо). Рішення про зарахування здобувачу результатів (певного освітнього компонента в цілому, або ж окремого виду навчальної роботи за таким освітнім компонентом) приймається уповноваженою Комісією з визнання результатів навчання за процедурою, визначеною вищезазначеним положенням.

12. Критерії, форми поточного та підсумкового контролю

Підсумковий бал (за 100-бальною шкалою) з дисципліни "Об'єктно-орієнтоване програмування" визначається як середньозважена величина, в залежності від питомої ваги кожної складової залікового кредиту:

Модуль 1		Модуль 2		Модуль 3	Модуль 4	Модуль 5
10 %	10%	10 %	10%	5 %	15 %	40%
Поточне оцінювання	Модульний контроль 1	Поточне оцінювання	Модульний контроль 2	Тренінг	Самостійна робота	Екзамен
Оцінка визначається як середнє арифметичне з отриманих оцінок за перший змістовий модуль (теми 1-6)	Виконання модульного завдання, складається із однієї практичної задачі (теми 1-6)	Оцінка визначається як середнє арифметичне з отриманих оцінок за другий змістовий модуль (теми 7-13)	Виконання модульного завдання, складається із однієї практичної задачі (теми 7-13)	Оцінка за виконання завдання	Оцінка за виконання самостійного завдання	Два теоретичні запитання (по 30 балів), одне практичне завдання (40 балів)

Поточне опитування під час заняття:

90–100 балів – у повному обсязі володіє навчальним матеріалом, вільно самостійно та аргументовано його викладає під час відповідей, глибоко та всебічно розкриває зміст теоретичних питань;

75–89 балів – достатньо повно володіє навчальним матеріалом, але при викладанні деяких питань не вистачає достатньої глибини та аргументації, допускаються при цьому окремі несуттєві неточності та незначні помилки;

65–74 бали – в цілому володіє навчальним матеріалом та викладає його основний зміст, але без глибокого всебічного аналізу, обґрунтування та аргументації, допускаючи при цьому окремі суттєві неточності та помилки;

60–64 бали – не в повному обсязі володіє навчальним матеріалом, фрагментарно (без аргументації та обґрунтування) його викладає, недостатньо розкриває зміст теоретичних питань, допускаючи при цьому суттєві неточності;

1–59 балів – не володіє навчальним матеріалом, не розкриває зміст теоретичних питань.

Підсумкова оцінка за поточне опитування кожного модуля визначається як середнє арифметичне оцінок, отриманих під час занять в межах кожного модуля.

Тренінг:

90–100 балів – у повному обсязі володіє навчальним матеріалом, вільно самостійно та аргументовано його використовує під час виконання завдань тренінгу, виявляє творчий підхід до виконання завдань;

75–89 балів – достатньо повно володіє навчальним матеріалом, але при виконанні окремих завдань тренінгу не вистачає достатньої глибини та аргументації його використання, допускаються при цьому окремі несуттєві неточності та незначні помилки, загалом виявляє творчий підхід до виконання завдань;

65–74 бали – в цілому володіє навчальним матеріалом та загалом його використовує при виконанні завдань тренінгу, але без глибокого всебічного аналізу, обґрунтування та аргументації, допускаючи при цьому суттєві неточності та помилки, в окремих моментах виявляє творчий підхід до виконання завдань;

60–64 бали – не в повному обсязі володіє навчальним матеріалом, фрагментарно (без аргументації та обґрунтування) його використовує, недостатньо розкриває зміст завдань тренінгу, допускаючи при цьому суттєві неточності, не виявляє творчого підходу до виконання завдань;

1–59 – не володіє навчальним матеріалом, не розкриває зміст завдань тренінгу, не бере участі у колективних завданнях під час проведення тренінгу.

Самостійна робота:

90–100 балів – зміст самостійної роботи повністю відповідає встановленим вимогам, містить елементи самостійного дослідження, свідчить про високий рівень опанування навчального матеріалу, студент на високому рівні виявляє творчий підхід до виконання завдань;

75–89 балів – зміст самостійної роботи в основному відповідає встановленим вимогам, можуть бути несуттєві недопрацювання за окремими завданнями, свідчить про належний рівень опанування навчального матеріалу, студент належно виявляє творчий підхід до виконання завдань;

60–74 балів – поставлені завдання виконані на недостатньому рівні; наведені авторські напрацювання є загальними і слабо обґрунтованими, свідчать про недостатній рівень опанування навчального матеріалу; студент припускається значних помилок у виконанні завдань, в окремих моментах виявляє творчий підхід до виконання завдань;

1–59 балів – завдання практично не виконані; відсутні авторські напрацювання; грубі помилки у вирішенні завдань роботи, що свідчать про низький рівень опанування навчального матеріалу; студент не виявляє творчого підходу до виконання завдань.

Модульна робота, екзамен – види контролю, при яких засвоєння студентом теоретичного та практичного матеріалу оцінюється від 0 до 100 балів як сума балів за виконані завдання. Екзаменаційний білет складається із:

задачі (практичного завдання): 1 задача, за розв’язання якої студент може отримати від 0 до 40 балів;

15–25 балів – у повному обсязі володіє навчальним матеріалом, правильно розв’язує практичне завдання і інтерпретує отримані результати, демонструє самостійність виконання.

5–14 балів – у достатньому обсязі володіє навчальним матеріалом, правильно розв’язує практичне завдання, але на додаткові контрольні запитання відсутня повна відповідь, допускає несуттєві неточності та фрагментарно (без аргументації) інтерпретує отримані результати, демонструє самостійність виконання.

1–4 балів – не в повному обсязі володіє матеріалом, фрагментарно розв’язує практичне завдання, допускає суттєві неточності, поверхнево його викладає, недостатньо розкриває зміст поставлених питань.

теоретичних (проблемних) питань: 2 теоретичні питання, за відповідь на кожне з них студент може отримати від 0 до 30 балів.

15–30 балів – у повному обсязі володіє навчальним матеріалом, всебічно, самостійно та аргументовано його викладає під час відповіді, глибоко та всебічно розкриває зміст завдання.

1–14 балів – в цілому володіє навчальним матеріалом, але не в повному обсязі, фрагментарно (без аргументації та обґрунтування) його викладає, недостатньо розкриває зміст завдання, допускаючи при цьому суттєві неточності, відповіді на запитання нечіткі.

Шкала оцінювання

За шкалою університету	За національною шкалою	За шкалою ECTS
90-100	Відмінно	A (відмінно)
85-89	Добре	B (дуже добре)
75-84		C (добре)
65-74	Задовільно	D (задовільно)
60-64		E (достатньо)
35-59	Незадовільно	FX (незадовільно з можливістю повторного складання)
1-34		F (незадовільно з обов’язковим повторним курсом)

13. Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

№	Найменування	Номер теми
1	IntelliJ IDEA	1-13
2	PyCharm Community Edition	1-13
3	Kotlin Playground	7-9
4	draw.io / diagrams.net	1-13
5	Android Studio	7-9
6	Swift Playgrounds	10-12
7	Xcode	10-12
8	Git + GitHub	1-13

Рекомендовані джерела інформації

1. Бублик В.В. Об'єктно-орієнтоване програмування: підручник. Київ: ІТ-книга, 2015. 624 с.
2. Бурлаков А.А. Об'єктно-орієнтований аналіз і проектування: метод. реком. Хмельницький: ХНУ, 2017. 136 с.
3. Гришанович Т. О., Глинчук Л. Я. Основи об'єктно-орієнтованого програмування : навч. посібник Луцьк : ВНУ імені Лесі Українки, 2022. 120 с.
4. Коноваленко І.В. Програмування мовою C# 6.0: навч. посіб. Тернопіль: ТНТУ. 2016. 227 с.
5. Кузьма К.Т. Програмування мобільних пристроїв: навчальний посібник для дистанційного навчання / К.Т. Кузьма. – Миколаїв: СПД Румянцева Г.В., 2021. 128 с.
6. Мнушка О.В., Савченко В.М., Маций О.Б. Об'єктно-орієнтоване програмування мовою Python: навчальний посібник для студентів напрямів підготовки 122 Комп'ютерні науки та 121 Інженерія програмного забезпечення. Харків: ХНАДУ, 2021. 228 с.
7. Стахіра Р. Й., Кулик П. Р., Шувар Р. Я. Kotlin для роботи з даними : навч. посіб. Львів, 2023. 160 с.