

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра програмних систем і технологій**

«ЗАТВЕРДЖУЮ»

Заступник декана

з навчально-виховної роботи

Наталія ТМЄНОВА

«29» серпня 2026 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

«ПРОГРАМУВАННЯ ПІД МОБІЛЬНІ ПЛАТФОРМИ»

для здобувачів вищої освіти

галузь знань	12 Інформаційні технології
спеціальність	121 Інженерія програмного забезпечення
освітній рівень	бакалавр
освітня програма	Інженерія програмного забезпечення
вид дисципліни	вибіркова

Форма навчання	денна
Навчальний рік	2026/2027
Семестр	8
Кількість кредитів ECTS	6
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладачі: Петрівський Володимир Ярославович, доктор філософії

Пролонговано: на 2027/2028 н.р. _____ (підпис) _____ (ПІБ) «___» _____ 2027 р. (дата)

на 2028/2029 н.р. _____ (підпис) _____ (ПІБ) «___» _____ 2028 р. (дата)

на 2029/2030 н.р. _____ (підпис) _____ (ПІБ) «___» _____ 2029 р. (дата)

КИЇВ – 2026

Розробник: **Петрівський Володимир Ярославович**, доктор філософії,
асистент кафедри програмних систем і технологій

Завідувач кафедри
програмних систем і технологій


(підпис) проф. Олексій БИЧКОВ

Протокол № 18 від 17 червня 2026 року

Схвалено науково-методичною комісією факультету інформаційних
технологій

Протокол № 11 від 26 червня 2026 року

Голова науково-методичної комісії 
(підпис) Ганна КРАСОВСЬКА

© Петрівський В.Я., 2026 рік
© _____, 20__ рік
© _____, 20__ рік

1. Мета дисципліни – закріплення навичок проектування застосунків і розробки інтерфейсу користувача в контексті створення мобільних додатків, отримання теоретичної і практичної підготовки в області розробки програм для платформ, що динамічно розвиваються і змінюються.

2. Попередні вимоги до опанування або вибору навчальної дисципліни:

1. Знати: методи конструювання програмного забезпечення, теоретичні основи мов програмування C++, C#, Java, JavaScript, основи проектування баз даних.
2. Вміти проектувати консольні та веб додатки, використання RESTful API, створювати архітектуру БД.
3. Володіти елементарними навичками програмування та використання баз даних.

3. Анотація навчальної дисципліни: дисципліна включає вивчення основ і отримання практичних навичок програмної інженерії в області розробки програмного забезпечення для мобільних пристроїв, ознайомлення з основними принципами і технологіями розробки додатків для мобільних операційних систем.

4. Завдання (навчальні цілі) – формування у студентів знання про принципи організації інтерфейсу користувача, архітектуру мобільного застосунку, навичок розробки і налагодження застосунків, що використовують можливості мобільних пристроїв, вміння створення багатоплатформових і енергоефективних застосунків. Здатність розв'язувати складні спеціалізовані завдання або практичні проблеми інженерії програмного забезпечення, що характеризуються комплексністю із застосуванням теорій та методів інформаційних технологій.

Набуття загальних компетентностей (ЗК):

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК3. Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК5. Здатність вчитися та оволодівати сучасними знаннями та проводити самостійні дослідження на відповідному рівні.

ЗК6. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

Набуття фахових компетентностей (ФК):

ФК2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

ФК3. Здатність розробляти архітектуру, модулі та компоненти програмних систем.

ФК4. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним

завданням та стандартами.

ФК5. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.

ФК6. Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки.

ФК7. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.

ФК8. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

ФК10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

ФК11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

ФК12. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності загальної функціональності і надійності програмного забезпечення.

ФК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

ФК14. Здатність до алгоритмічного та логічного мислення.

5. Результати навчання за дисципліною: (описуються з детальною достовірністю для розробки заходів оцінювання)

Результат навчання		Методи викладання і навчання	Методи оцінювання	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
1.1	Знати рівень ядра, рівень бібліотек, рівень додатків, Android SDK, архітектуру застосунку	Лекція, Лабораторне заняття	Звіт про виконану роботу у формі співбесіди	7,69%
1.2	Знати шаблони і технологію створення користувацького інтерфейсу мобільного застосунку	Лекція, Лабораторне заняття	Звіт про виконану роботу у формі співбесіди	7,69%
1.3	Знати поняття діалогу, типи діалогових вікон, етапи створення меню і діалогових вікон	Лекція, Лабораторне заняття	Звіт про виконану роботу у формі співбесіди	7,69%
1.4.	Знати компоненти мобільного застосунку, їх життєвий цикл, особливості взаємодії між собою	Лекція, Лабораторне заняття	Звіт про виконану роботу у формі співбесіди	7,69%
1.5	Знати особливості взаємодії з апаратними засобами	Лекція, Лабораторне заняття	Звіт про виконану роботу у формі співбесіди	7,69%
1.6	Знати технологію роботи з базами даних, файлами і ресурсами	Лекція, Лабораторне заняття	Звіт про виконану роботу у формі співбесіди	7,69%
2.1	Вміти використовувати різні типи ресурсів, працювати з ресурсами в середовищі Android studio	Лекція, Лабораторне заняття	Звіт про виконану роботу у формі співбесіди	7,69%

2.2	Вміти розробляти користувацький інтерфейс, використовувати різні версії дизайну	Лекція, Лабораторне заняття	Звіт про виконану роботу у формі співбесіди	7,69%
2.3	Вміти створювати і використовувати колекції в додатках	Лекція, Лабораторне заняття	Звіт про виконану роботу у формі співбесіди	7,69%
2.4	Вміти створювати і використовувати класи активності і наміру	Лекція, Лабораторне заняття	Звіт про виконану роботу у формі співбесіди	7,69%
2.5	Вміти зберігати основний і проміжний стани Activity, захищені файли довільного формату в пам'яті пристрою External Storage, загальні файли довільного формату в зовнішній пам'яті (SD Card)	Лекція, Лабораторне заняття	Звіт про виконану роботу у формі співбесіди	7,69%
2.6	Вміти створювати графічні елементи мобільних додатків, працювати з аудіо і відео записами	Лекція, Лабораторне заняття	Звіт про виконану роботу у формі співбесіди	7,69%
2.7	Вміти створювати, відкривати і закривати бази даних, створювати таблиці	Лекція, Лабораторне заняття	Звіт про виконану роботу у формі співбесіди	7,69%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання (необов'язково для вибірових дисциплін)

Результати навчання дисципліни	1.1	1.2	1.3	1.4	1.5	1.6	2.1	2.2	2.3	2.4	2.5	2.6	2.7
Програмні результати навчання													
ПР-12. Застосовувати на практиці ефективні підходи щодо проєктування програмного забезпечення.	+		+	+		+		+	+		+		+
ПР-30. Знати, розуміти, вміти застосовувати моделі математичної теорії ігор та методи розробки алгоритмів ігрових програм із застосуванням штучного інтелекту для розробки програмного забезпечення комп'ютерних ігор, вміти застосовувати методи цифрового синтезу і обробки зорового контенту, створювати, перетворювати, оцифровувати, обробляти і виводити зображення засобами обчислювальної техніки, а також володіти інструментами та технологіями графічного дизайну, вміти творчо вирішувати композиційні, стилістичні та комерційні завдання проєкту.	+	+	+		+		+			+		+	+

7. Схема формування оцінки

Оцінювання проводиться відповідно до «Рекомендацій щодо оцінювання освітніх STEM-проєктів», затверджених на засіданні кафедри програмних систем і технологій (протокол № 16 від «01» травня 2026 р.):

<https://pst.knu.ua/academics/bachelor/evaluation-stem-projects/>.

7.1 Проєктна складова оцінки

Оцінка за проєкт – це оцінка студента за роботу в командному проєкті, а

не оцінка команди.

Тому бали розділені на дві складові:

- Командна складова: 30 балів проєкту (блок С, технічна/математична якість продукту; у семестрі → 12 балів). Команда створила один продукт, тож його якість – спільна відповідальність і оцінюється однаково для всіх членів команди.

- Індивідуальна складова: 70 балів проєкту (блоки А + В + D + E: процес, виконання власної зони відповідальності, складність цієї зони та захист; у семестрі → 28 балів). Ці бали індивідуальні в кожного студента.

Формула для кожного студента:

Оцінка_i = Продукт_команди (спільне) + Індивідуальне_i.

У таблиці представлено розподіл ваг та балів для кожного блоку.

Код	Критерій	Вага	Об'єктивне правило нарахування рівня (у %, 0–100)
А. Процес і дисципліна (індивід.)			
A1	Дотримання проміжних дедлайнів (3 чекпоінти)	6	Рівень за дискретною шкалою (не частка): усі 3 чекпоінти вчасно з повним переліком артефактів → 1.0; 1 етап із затримкою ≤24 год → 0.8; 2 з 3 вчасно → 0.6; 1 з 3 → 0.4; жодного або постфактум → 0.
A2	Ведення репозиторію / логбука	5	+0.2 за кожен виконаний пункт (перевірка Git-історії): (1) ≥1 змістовний коміт/тиждень; (2) коміти прив'язані до issue; (3) README з інструкцією запуску; (4) лог рішень ≥5 записів «проблема→рішення»; (5) гілкова модель (не все в main).
A3	Опрацювання зворотного зв'язку	4	Рівень = частка зауважень із чекпоінтів, оформлених як issue і закритих до наступного етапу. 1.0 ≥90%; 0.8 = 70–89%; 0.6 = 50–69%; 0.4 = 30–49%; 0 = <30%.
В. Внесок і надійність (індивід.)			
B1	Виконання індивідуальної зони відповідальності	12	Зона фіксується на Зрізі 0. Рівень оцінюється за сукупністю артефактів (ланцюг issue → branch → PR → коміти → review → merge; код, архітектурні рішення, тести, документація, debugging); story points — лише один із доказів, не самоціль. Повний обсяг якісно → 1.0; переважно (незначні прогалини) → 0.8; частково → 0.6; фрагментарно → 0.4; мінімально → 0.2; не виконано → 0. Авторство визначається за виконаною роботою, а не git blame.
B2	Виконання зобов'язань і комунікація ризиків	8	Рівень = частка взятих задач, які АБО виконані в узгоджений строк, АБО за якими ризик/затримку завчасно (до дедлайну) комуніковано з оновленням issue та новою датою. 1.0 ≥95%; 0.8 = 80–94%; 0.6 = 60–79%; 0.4 = 40–59%; 0.2 = 20–39%; 0 = <20%. Мовчазний зрив дедлайну як виконане не зараховується.

B3	Результативність code-review	5	Зараховуються лише результативні рев'ю (провів рев'ю чужого PR → знайшов суттєву проблему або запропонував обґрунтоване покращення → автор відреагував → перевірено виправлення). ≥ 3 результативних → 1.0; 2 → 0.8; 1 → 0.5; 0 → 0. Кількість дрібних косметичних коментарів не враховується.
С. Технічна якість продукту (КОМАНДНЕ — однакове всім)			
C1	Функціональна коректність (приймальні кейси)	12	Рівень = пройдено приймальні тест-кейси / усього узгоджених кейсів (фіксований список кейсів). Прямий розрахунок частки.
C2	Обробка крайніх випадків і помилок	5	+0.2 за кожен: (1) валідація введення; (2) обробка порожніх/некоректних даних; (3) обробка винятків без аварійного завершення; (4) межові значення; (5) відсутність необроблених винятків у лог-файлі прогону.
C3	Автотести і покриття	6	1.0 = автотести + покриття $\geq 80\%$ (звіт інструмента) + СІ гаяє тести; 0.8 = автотести + покриття $\geq 60\%$; 0.6 = автотести + покриття $\geq 40\%$; 0.4 = автотести є, покриття $< 40\%$; 0 = автотестів немає.
C4	Якість коду за статичним аналізом	4	Critical/major зауважень лінера-аналізатора на 1000 рядків: 0 → 1.0; ≤ 2 → 0.8; ≤ 5 → 0.6; ≤ 10 → 0.4; > 10 → 0. Інструментальна перевірка (pylint/ESLint/SonarQube).
C5	Цілісність і безпека роботи з даними / схеми БД	3	+0.2 за кожен застосовний пункт: (1) нормалізація ≥ 3 НФ або обґрунтована денормалізація; (2) РК/ФК та обмеження цілісності; (3) індекси на полях пошуку/JOIN; (4) параметризовані запити (захист від ін'єкцій); (5) транзакції де потрібно. Для проєктів без БД — еквівалент для файлових даних (валідація, транзакційність, узгодженість).
Д. Складність зони відповідальності (індивід.)			
D1	Складність і повнота індивідуальної зони відповідальності	10	Зона фіксується на Зрізі 0 (для індивідуальних проєктів — весь проєкт). Рівень = складність зони × повнота якісного виконання: експертна зона повністю виконана (event-driven backend, складна авторизація, розподілені воркери тощо) → 1.0; просунута → 0.9; базова, повністю та якісно виконана → 0.8; зона виконана частково/спрощено → 0.4–0.5; тривіальний внесок (напр. README + шаблонний CRUD) → 0.2; не виконано → 0. Добре виконаний базовий рівень НЕ карається.
Е. Захист і розуміння (індивід.)			
E1	Відповіді за відкритим банком питань	12	Рівень = сума за відповіді / максимум. Студент отримує К питань з оприлюдненого банку; кожна відповідь: повна = 1, часткова = 0.5, немає = 0.
E2	Модифікація умови «на льоту»	5	2 заздалегідь заготовлені мікрозміни умови; кожна коректно пояснена/реалізована = 0.5. Рівень = 0 / 0.5 / 1.0.

ЕЗ	Демонстрація запуску за README	3	Запускається з нуля за README без втручання автора у визначений ліміт часу → 1.0; із дрібним втручанням → 0.5; не запускається → 0.
РАЗОМ (сума ваг)		100	Підсумок = MIN(100; М × Σ(вага×рівень) + Бонус)

7.1.1 Академічна доброчесність і політика щодо ШІ

Доброчесність застосовується як зовнішній множник М до підсумкового бала – Оцінка = М × MIN(100; Σ + Бонус):

М = 1.0 – порушень не виявлено.

М = 0.5 – встановлене порушення: плагіат, запозичення коду/викладок без атрибуції, недеклароване використання згенерованого коду. Застосовується за рішенням згідно з політикою кафедри.

М = 0.0 – груба фальсифікація результатів або видавання чужої роботи за власну; підсумок = 0 (навіть із бонусом), робота передається на розгляд комісії.

Використання інструментів метапрограмування в широкому сенсі, зокрема ШІ-інструментів генерації коду та інших інструментів, пов'язаних з генерацією і автоматичною обробкою текстів формальними і неформальними мовами в контексті процесу розробки ПЗ у процесі виконання проєктної роботи є дозволеним (і, залежно від завдання роботи, може бути рекомендованим) за таких обов'язкових умов:

(1) у файлі META-USAGE студент декларує застосовані принципи (мета)програмування, їх переваги і обмеження у контексті проєктної роботи, а також конкретні застосовані інструменти, ключові запити до них, згенеровані фрагменти коду / інші артефакти;

(2) на захисті студент демонструє володіння матеріалом файлу META-USAGE, зокрема, здатний відповісти на додаткові питання щодо обраної стратегії застосування згаданих інструментів та її порівняння з альтернативними стратегіями, що впливають з раніше пройденого матеріалу даної дисципліни та з попередніх дисциплін;

(3) на захисті студент демонструє розуміння змісту згенерованих фрагментів коду / артефактів, способів перевірки їх якостей (наприклад, коректності, ефективності і т.д.), можливих шляхів їх подальшого використання.

Недекларований згенерований результат кваліфікується як порушення доброчесності (див. вище). Нездатність пояснити власний результат безпосередньо знижує бали блоку Е (захист) за відповідними правилами.

7.2 Індивідуальна складова оцінки

Індивідуальна оцінка обчислюється як сума балів за індивідуальні лабораторні роботи та балів за модульні контрольні роботи.

Розподіл балів за видами індивідуальної діяльності:

Вид діяльності	Кількість балів
Лабораторні роботи	10
Модульна контрольна робота 1	5
Модульна контрольна робота 2	5
Разом	20

Теми лабораторних робіт:

Тема лабораторної роботи	Кількість балів
ЛР1. Створення базового мобільного застосунку та структури проєкту. Розроблення користувацького інтерфейсу мобільного застосунку.	2
ЛР2. Реалізація логіки застосунку та обробки подій. Навігація між екранами та передача даних.	2
ЛР3. Робота зі списками та колекціями даних.	2
ЛР4. Локальне збереження та відновлення даних.	2
ЛР5. Використання дозволів та можливостей мобільного пристрою.	2
Разом	10

7.3 Підсумкове оцінювання (іспит):

- умови допуску до іспиту:

Студенти, які в семестрі набрали **менше 36 балів** до екзамену **не допускаються**. Для допуску до екзамену треба обов'язково доздати теми лабораторних занять, за якими результати семестрового оцінювання відсутні або здати особистий проєкт мобільного застосунку, в якому відображені питання 8 тем лабораторних занять.

	Робота під час семестру	Іспит	Підсумкова оцінка
Мінімум	36	24	60
Максимум	60	40	100

Оцінка за іспит не може бути меншою **24 балів** для отримання загальної позитивної оцінки за курс.

Загальна сума балів семестрового і підсумкового оцінювання **не може перевищувати 100 балів**.

7.4. Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій і лабораторних занять

№ п/ п	Назва теми	Кількість годин		
		лекції	Лабораторні заняття	С/Р
Змістовний модуль 1.				
1	Тема 1. Операційні системи для мобільних пристроїв	2	2	12
2	Тема 2. Операційна система Android	2	4	12
3	Тема 3. Операційна система iOS	2	4	12
4	Тема 4. Архітектурні патерни при розробці мобільних додатків	4	6	24
	Модульна контрольна робота 1		2	
Змістовний модуль 2.				
6	Тема 5. Мова програмування Kotlin	2	4	12
7	Тема 6. Мова програмування Swift	2	4	12
8	Тема 7. React Native	2	4	12
9	Тема 8. Розробка мобільного додатку з використанням Kotlin	2	2	12
	Тема 9. Розробка мобільного додатку з використанням React Native	2	2	12
	Модульна контрольна робота 2		2	
	Письмовий екзамен			
	ВСЬОГО	20	36	120

Загальний обсяг 180 год., в тому числі:

лекцій – 20 год.;
 лабораторні заняття – 36 год.;
 консультація – 4 год.;
 самостійна робота – 120 год.

Рекомендована література

1. Gauchat J.D. SwiftUI for Masterminds 2026: How to take advantage of SwiftUI and iOS 26 to create insanely great apps for iPhones, iPads and Macs. – Mind Books, 2026. – ISBN 1069994200
2. Horton J. Android Programming for Beginners / John Horton. – Packt Publishing, 2022. – ISBN 978-1803236969
3. The Swift Programming Language. – [Електронний ресурс]. – Режим доступу: <https://docs.swift.org/swift-book/documentation/the-swift-programming-language/>
4. Hardy, K. A., Phillips, B., Marsicano, C. B. Android Programming: The Big Nerd Ranch Guide. Big Nerd Ranch, 2023.
5. Us Sunnat S. M. Building Mobile Apps with Flutter / S. M. Mohi Us Sunnat. – Packt Publishing, 2022. – ISBN 978-1801818402

6. Biessek A. Flutter for Beginners / Alessandro Biessek. – Packt Publishing, 2022. – ISBN 978-1800563685
7. Apple Inc. Education. Develop in Swift Explorations. Apple Books, 2024
8. Moskala M., Wojda I. Android Development with Kotlin / Marcin Moskala, Igor Wojda. – Packt Publishing, 2022. – ISBN 978-1803235832
9. Кузьма К. Т. Програмування мобільних пристроїв: навчальний посібник для дистанційного навчання / К.Т. Кузьма. – Миколаїв: СПД Румянцева Г. В., 2021. – 128 с. ISBN 567-859-837-548-4.
10. Портал розробників на Android. – [Електронний ресурс]. – Режим доступу: <https://developer.android.com/guide>
11. Портал розробників на iOS. – [Електронний ресурс]. – Режим доступу: <https://developer.apple.com/develop/>
12. Портал розробників на C# (Xamarin). – [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/en-us/xamarin/>
13. Ткаченко О. А. Розробка мобільних додатків під Android : навч. посіб. / Олександр Андрійович Ткаченко, Костянтин Олександрович Ткаченко, Олена Антонівна Чайковська ; Київ. нац. ун-т культури і мистецтв. – Київ : КНУКіМ, 2017. – 274 с. : іл
14. Розробка мобільних Дворецький М. Л., Нездолій Ю. О., Дворецька С. В., Кандиба І. О. Розробка мобільних застосунків для OS Android : навч. посіб. – Миколаїв : Вид-во ЧНУ ім. Петра Могили.