


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЕКОНОМІКО-ТЕХНОЛОГІЧНИЙ ІНСТИТУТ ІМЕНІ Р.ЕЛЬВОРТИ

Кафедра інформаційних технологій

"ЗАТВЕРДЖУЮ"
Завідувач кафедри

 /О.П. Бондар/
30 серпня 2024 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

«КРОС-ПЛАТФОРМНЕ ПРОГРАМУВАННЯ»

Спеціальність: 122 Комп'ютерні науки

Освітньо-професійна програма: Комп'ютерні науки

Нормовані дані	Курс	Семестр	Всього годин за планом	Кількість кредитів ECTS	Всього аудит (год.)	Аудиторних годин, (у тому числі КЗ)			Самостійна робота (год.)	Курсове проектування (семестр/кредити)	Контрольний підсумок (семестр)	
						Лекції	Лабораторні роботи	Практичні заняття			Екзамен	Залік
Форма навчання												
Денна	3	5	120	4	72	36		36	48		5	

Робочу програму складено на основі освітньо-професійної програми за спеціальністю: 122 Комп'ютерні науки

Робочу програму складено: Ізвалов О.В.

Робочу програму затверджено на засіданні кафедри інформаційних технологій.

Протокол № 1 від "30" серпня 2024 року

Завідувач кафедри  /О.П. Бондар/

Схвалено Вченою радою ЕТІ ім. Р.Ельворті

Протокол № 15 від "24" вересня 2024 року

Голова Вченої ради  /Штець Т.Ф. /

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни	
		Денна форма навчання	Заочна форма навчання
Кількість кредитів – 4	Галузь: 12– Комп’ютерні науки Спеціальність або освітня програма 122 Комп’ютерні науки	Статус дисципліни нормативна	
Змістових модулів - 4		Рік підготовки	
Індивідуальне завдання студента		3	
Загальна кількість годин -120		Семестр	
Тижневих годин для денної форми навчання: аудиторних: 1 семестр – 4 години	Ступінь вищої освіти: бакалавр	5	
		Лекції (год.)	
		36	
		Практичні, семінарські (год.)	
		36	
		Лабораторні (год.)	
		-	
		Самостійна робота (год.)	
		48	
		Індивідуальне завдання (год.)	
-			
-			
Вид контролю:			
5 сем – екзамен			

Примітка.

Співвідношення кількості годин аудиторних занять та самостійної роботи становить (%): 60% до 40% .

2. Мета та завдання навчальної дисципліни

2.1. МЕТА ВИКЛАДАННЯ ДИСЦИПЛІНИ

Кросплатформене програмування є однією з головних засад сучасної комерційної розробки. Оволодіння технологією побудови програмного забезпечення для широкого спектру апаратних та програмних платформ є важливою передумовою затребуваного спеціаліста.

Метою викладання дисципліни є надання студентам комплексних знань та практичних навичок у галузі кросплатформного програмування, спрямованих на розробку програмного забезпечення, яке може ефективно функціонувати на різних платформах та пристроях. Здобуття практичного досвіду розробки крос-платформних додатків через участь у проектах та вирішення завдань реального життя.

2.2. ЗАВДАННЯ ВИВЧЕННЯ ДИСЦИПЛІНИ

Вивчення теоретичних понять кросплатформеного програмування та ознайомлення з основними розповсюдженими технологіями, опанування базових методів досягання кросплатформеності та адаптації програмних продуктів під різні операційні системи та форм-фактори пристроїв.

2.3. За результатами вивчення дисципліни здобувач повинен опанувати наступні компетентності:

Інтегральна компетентність. Здатність розв’язувати складні спеціалізовані задачі та

практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.

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

ЗК6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

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.

СК18. Здатність проектувати та прототипувати інтерфейси програмних продуктів, використовуючи засоби комп'ютерної графіки.

2.4. За результатами вивчення навчальної дисципліни студент повинен:

знати:

- Основи кросплатформеної розробки та її переваги.
- Особливості роботи з мобільними платформами (iOS, Android).
- Архітектурні принципи побудови кросплатформених додатків.
- Основи мови програмування для кросплатформеного середовища.
- Поняття UI/UX у контексті кросплатформеності.
- Способи взаємодії з API різних платформ.
- Інструменти для тестування та налагодження додатків.

вміти:

- Створювати кросплатформені додатки для мобільних пристроїв.
- Використовувати сучасні фреймворки (Flutter, Cordova, AIR).
- Налаштовувати адаптивний інтерфейс для різних платформ.
- Інтегрувати API у кросплатформені проекти.
- Виконувати тестування та дебагінг додатків.
- Оптимізувати продуктивність кросплатформених додатків.
- Розгортати додатки у маркетплейсах (App Store, Google Play).

2.5. Програмні результати навчання

ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

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.

ПР19. Застосовувати засоби обробки графічних зображень для проектування інтерфейсів програмних продуктів.

2.6. Структурно-логічне місце дисципліни в освітній програмі

Попередні дисципліни:	Наступні дисципліни:
Програмування	Виробнича практика
Об'єктно-орієнтоване програмування	Технології комп'ютерного проектування
Веб-технології та веб-дизайн	Виконання кваліфікаційної роботи

Програма навчальної дисципліни складається з чотирьох модулів

3. Програма навчальної дисципліни

Змістовий Модуль 1. Принципи кросплатформеності та її реалізація

Тема 1.1. Основи кросплатформеного програмування

1. Переваги та недоліки кросплатформеного підходу.
2. Типи кросплатформених середовищ.
3. Відмінності між нативними та кросплатформеними додатками.

Тема 1.2. Архітектура кросплатформених додатків

1. Багатопарова архітектура додатків.
2. Організація коду для повторного використання.
3. Способи оптимізації архітектури під різні платформи.

Тема 1.3. Процеси розробки у кросплатформеному середовищі

1. Особливості роботи з IDE для кросплатформеності.
2. Підхід Continuous Integration/Continuous Deployment.
3. Роль тестування в циклі розробки.

Змістовий Модуль 2. Інструментарій побудови кросплатформених додатків

Тема 2.1. Огляд популярних фреймворків

1. Особливості роботи у Flutter: виджети та продуктивність.
2. Використання Cordova для веб-орієнтованих додатків.
3. Harman AIR: інтеграція мультимедіа у кросплатформених проектах.

Тема 2.2. Робота з інтерфейсами у кросплатформених середовищах

1. Побудова адаптивних UI-інтерфейсів.
2. Використання компонентів бібліотек фреймворків.
3. Анімації та їх підтримка на різних платформах.

Тема 2.3. Взаємодія з платформеними API

1. Підключення нативних модулів до кросплатформеного коду.
2. Робота з геолокацією, камерою та сенсорами.
3. Організація мережевої взаємодії (REST, GraphQL).

Змістовий Модуль 3. Особливості публікації мобільних додатків

Тема 3.1. Підготовка додатків до релізу

1. Оптимізація продуктивності та розміру додатка.
2. Робота з конфігураційними файлами.
3. Налаштування безпеки додатка.

Тема 3.2. Публікація додатків у Google Play Store

1. Реєстрація розробника та налаштування консолі.
2. Підготовка до публікації: APK або AAB.
3. Вимоги до дизайну сторінки додатка.

Тема 3.3. Публікація додатків в App Store

1. Налаштування профілю розробника.
2. Робота з Xcode для створення IPA-файлу.
3. Процедура проходження App Review.

Тема 3.4. Розробка оновлень і підтримка додатків

1. Створення змін у кросплатформеному середовищі.
2. Автоматизація оновлень через інструменти CI/CD.
3. Взаємодія з користувачами для виправлення помилок.

Змістовий Модуль 4. Скриптове кросплатформне програмування

Тема 4.1. Вступ до скриптового програмування

1. Роль скриптів у кросплатформеній розробці.

2. Використання Python і JavaScript для кросплатформеності.
3. Основи створення скриптів для автоматизації.

Тема 4.2. Створення утиліт для мобільних додатків

1. Автоматизація збору даних через скрипти.
2. Створення кросплатформених інструментів CLI.
3. Застосування скриптів для управління API.

Тема 4.3. Інтеграція скриптів у додатки

1. Використання скриптів для обробки даних.
2. Вбудовування скриптів у кросплатформені середовища.
3. Автоматизація взаємодії між компонентами додатків.

Тема 4.4. Оптимізація та тестування скриптів

1. Тестування скриптів на різних платформах.
2. Методи оптимізації коду скриптів.
3. Впровадження логування та обробки помилок.

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	усього	у тому числі				
		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7
Змістовий модуль 1. Принципи кросплатформеності та її реалізація						
Тема 1.1. Основи кросплатформеного програмування	3	2				1
Тема 1.2. Архітектура кросплатформених додатків	11	2	4			5
Тема 1.3. Процеси розробки у кросплатформеному середовищі	16	5	5			6
Разом за змістовим модулем 1	30	9	9			12
Змістовий модуль 2. Інструментарій побудови кросплатформених додатків						
Тема 2.1. Огляд популярних фреймворків	3	2				1
Тема 2.2. Робота з інтерфейсами у кросплатформених середовищах	11	2	4			5
Тема 2.3. Взаємодія з платформеними API	16	5	5			6
Разом за змістовим модулем 2	30	9	9			12
Змістовий модуль 3. Особливості публікації мобільних додатків						
Тема 3.1. Підготовка додатків до релізу	3	2				1
Тема 3.2. Публікація додатків у Google Play Store	9	4	4			1
Тема 3.3. Публікація додатків в App Store	9	2	5			2
Тема 3.4. Розробка оновлень і підтримка додатків	9	1				8
Разом за змістовим модулем 3	30	9	9			12
Змістовий модуль 4. Скриптове кросплатформне програмування						

Тема 4.1. Вступ до скриптового програмування	3	2			1
Тема 4.2. Створення утиліт для мобільних додатків	9	4	4		1
Тема 4.3. Інтеграція скриптів у додатки	9	2	5		2
Тема 4.4. Оптимізація та тестування скриптів	9	1			8
Разом за змістовим модулем 4	30	9	9		12
Разом	120	36	36		48

5. Теми семінарських занять

№ з/п	Назва теми	Кількість годин
1	Не передбачено	

6. Теми практичних занять

№ з/п	Назва теми	Кількість годин
Змістовий модуль 1. Принципи кросплатформеності та її реалізація		9
1.	Архітектура кросплатформених додатків Багатошарова архітектура додатків. Організація коду для повторного використання. Способи оптимізації архітектури під різні платформи.	4
2.	Процеси розробки у кросплатформеному середовищі Особливості роботи з IDE для кросплатформеності. Підхід Continuous Integration/Continuous Deployment. Роль тестування в циклі розробки.	5
Змістовий модуль 2. Інструментарій побудови кросплатформених додатків		9
3.	Робота з інтерфейсами у кросплатформених середовищах Побудова адаптивних UI-інтерфейсів. Використання компонентів бібліотек фреймворків. Анімації та їх підтримка на різних платформах.	4
4.	Взаємодія з платформеними API Підключення нативних модулів до кросплатформеного коду. Робота з геолокацією, камерою та сенсорами. Організація мережевої взаємодії (REST, GraphQL). Змістовий Модуль 3: Особливості публікації мобільних додатків	5
Змістовий модуль 3. Особливості публікації мобільних додатків		9
5.	Публікація додатків у Google Play Store Реєстрація розробника та налаштування консолі. Підготовка до публікації: APK або AAB. Вимоги до дизайну сторінки додатка.	4
6.	Публікація додатків в App Store Налаштування профілю розробника. Робота з Xcode для створення IPA-файлу. Процедура проходження App Review	5
Змістовий модуль 4. Скриптове кросплатформне програмування		9
7.	Створення утиліт для мобільних додатків Автоматизація збору даних через скрипти.	4

	Створення кроссплатформених інструментів CLI. Застосування скриптів для управління API.	
8.	Інтеграція скриптів у додатки Використання скриптів для обробки даних. Вбудовування скриптів у кроссплатформені середовища. Автоматизація взаємодії між компонентами додатків.	5
	Разом	36

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Не передбачено	

8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Тема 3.4. Розробка оновлень і підтримка додатків 1. Створення змін у кроссплатформеному середовищі. 2. Автоматизація оновлень через інструменти CI/CD. 3. Взаємодія з користувачами для виправлення помилок.	8
2	Тема 4.4. Оптимізація та тестування скриптів 1. Тестування скриптів на різних платформах. 2. Методи оптимізації коду скриптів. 3. Впровадження логування та обробки помилок.	8
3	Опрацювання лекцій та підготовка до практичних занять	32
	Разом	48

9. Індивідуальні завдання

Не передбачено

10. Методи навчання

Лекції із застосування мультимедійних технологій навчання, практичні заняття для формування навичок розв'язування задач, консультації.

11. Методи контролю

Оцінювання якості знань студентів здійснюється шляхом поточного, підсумкового (семестрового) контролю за 100-бальною шкалою оцінювання, за шкалою ECTS та національною шкалою оцінювання.

Поточний контроль - оцінювання засвоєння студентом навчального матеріалу під час проведення практичних занять, виконання індивідуальних домашніх завдань, консультацій. Результати поточного контролю заносяться в журнал у балах (5, 4, 3, 2, 1). Сума балів за поточний контроль максимально дорівнює 60.

Підсумковий (семестровий) контроль – комплексне оцінювання якості засвоєння навчального матеріалу дисципліни на екзамені. Сума балів за екзамен максимально дорівнює 40.

12. Розподіл балів, які отримують студенти

Поточне оцінювання та самостійна робота						Екзамен	Сума
Змістовий модуль №1	Змістовий модуль №2	МКР	Змістовий модуль №3	Змістовий модуль №4	МКР		
10	10	10	10	10	10	40	100

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проєкту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

13. Методичне забезпечення

1. Навчально-методичний комплекс.
2. Відео лекцій (в електронному варіанті).

14. Рекомендована література

Основна

1. Філіпс Б. Стюарт К. Марсикано К. Android. Програмування для професіоналів. 2-е видання. Київ: Діалектика, 2021. 960 с.
2. Ткаченко О.М. Комп'ютерне програмування. Навчальний посібник. - К.: ФОП Ямчинський О.В., 2020. - 304 с.
3. Алгоритми і структури даних: навчальний посібник / Н. Б. Шаховська; Р.О. Голощук; за заг. ред. Пасічника В.В. - Львів: Магнолія 2006, 2011. - 215 с
4. Поморова О. В., Говорущенко Т. О. Проєктування інтерфейсів користувача: навч. посіб. Хмельницький: ХНУ. 2011. 206 с.
5. Кросплатформне програмування: конспект лекцій // Укладач Коротун Т.М. - К. : МНТУ.– 2017.

Допоміжна

1. Flutter Cookbook: 100+ step-by-step recipes for building cross-platform, professional-grade apps with Flutter 3.10.x and Dart 3.x, 2nd Edition – Packt Publishing – 2023 – 712 с.
2. Griffiths David, Griffiths Dawn. Head First Android Development. O'Reilly Media, Inc., 2022. – 1410 с
3. Cross-Platform GUI Programming with wxWidgets / Julian Smart, Kevin Hock, Stefan Csomor.: Prentice Hall, 2005. – 744 с.

15. Інформаційні ресурси

1. <https://flutter.dev/>
2. <https://dartpad.dev/>
3. <https://developers.google.com/learn/pathways/intro-to-flutter>
4. <https://cordova.apache.org/>
5. <https://airsdk.harman.com/>