

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

/назва факультету/

Кафедра комп'ютерних наук

/назва кафедри/



“ЗАТВЕРДЖУЮ”

Декан факультету

Ігор БАРАН

2024 року

“ 02 ”

109

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Крос-платформне програмування

/назва дисципліни/

галузь знань 12 Інформаційні технології

/шифр і назва галузі знань/

рівень вищої освіти Перший (бакалаврський рівень)

/назва/

спеціальність 122 «Комп'ютерні науки»

/шифр і назва/

освітня програма Комп'ютерні науки

/назва/

спеціалізація

/назва/

вид дисципліни вибіркова циклу професійної підготовки

/обов'язкова / вибіркова/

Робоча програма з навчальної дисципліни Крос-платформне програмування
/назва дисципліни/

для студентів факультету комп'ютерно-інформаційних систем і програмної інженерії
/назва факультету(ів)/

Розробник:

асистент кафедри комп'ютерних наук,

доктор філософії

/посада, науковий ступінь та вчене звання/

/підпис/

Олег ПАЛКА

/ініціали та прізвище/

Робоча програма розглянута та схвалена

на засіданні кафедри комп'ютерних наук
/назва/

Протокол від « 26 » серпня 2024 року № 1

Завідувач кафедри

/підпис/

Ігор БОДНАРЧУК

/ініціали та прізвище/

Робоча програма розглянута та схвалена НМК

факультету комп'ютерно-інформаційних систем і програмної інженерії

Протокол від « 2 » вересня 2024 року № 1

Секретар НМК

/підпис/

Богдана МЛИНКО

/ініціали та прізвище/

Робоча програма погоджена:

Спеціальність 122 «Комп'ютерні науки»

/шифр і назва/

освітня програма Комп'ютерні науки

/назва/

Завідувач випускової кафедри

/підпис/

Ігор БОДНАРЧУК

/ініціали та прізвище/

Гарант освітньої програми

/підпис/

Леся ДМИТРОЦА

/ініціали та прізвище/

1. Структура навчальної дисципліни

Показник	Всього годин	
	Денна форма навчання (ДФН)	Заочна (дистанційна) форма навчання (ЗФН)
Кількість кредитів/год	5/150	5/150
Аудиторні заняття, год.	54	12
Самостійна робота, год.	96	138
Аудиторні заняття:		
– лекції, год.	18	6
– лабораторні заняття, год.	36	6
– практичні заняття, год.	–	–
– семінарські заняття, год.	–	–
Самостійна робота:		
– підготовка до лабораторних (практичних – семінарських) занять, год.	24	24
– опрацювання окремих розділів програми, які не виносяться на лекції, год.	52	86
– виконання контрольних завдання, год.	–	–
– виконання індивідуальних завдань, год.	–	–
– виконання курсових проектів (робіт), год.	–	–
– підготовка та складання <u>заліків</u> , екзаменів, контрольних робіт, рефератів, есе, <u>тестування</u> , год.	20	28
Екзамен	–	–
Залік	3	3

Частка годин самостійної роботи студента:

денна форма навчання - 64 %;

заочна (дистанційна) форма навчання - 92 %.

2. Мета та завдання навчальної дисципліни

2.1. Мета вивчення навчальної дисципліни

Основною метою дисципліни «Крос-платформне програмування» є отримання студентами теоретичних знань та практичних навичок у сфері сучасних підходів, методів і технологій проектування, розробки, тестування та публікації крос-платформного програмного забезпечення. Особлива увага приділяється поглибленню практичного досвіду при використанні мови програмування C# та популярних фреймворків, таких як React Native і Flutter, у процесі створення програмних застосунків для різних платформ.

Предметом вивчення дисципліни “Крос-платформне програмування” є сучасні підходи, технології та інструменти розробки крос-платформного програмного забезпечення, включаючи проектування архітектури застосунків, створення адаптивних інтерфейсів користувача, роботу з базами даних, використання регулярних виразів, а також методи тестування, публікації та автоматизації процесів розгортання за допомогою CI/CD. Особлива увага приділяється практичному використанню мов програмування (C#, Dart, JavaScript) та популярних фреймворків (React Native, Flutter) у процесі створення програмних застосунків для різних платформ.

Важливим аспектом успішного засвоєння навчального матеріалу дисципліни є самостійна робота студентів з використанням інструментальних засобів розробки та тестування програмного забезпечення (інтегрованих середовищ розробки) та відповідної технічної літератури.

2.2. Завдання навчальної дисципліни

Завданням дисципліни є формування у студентів теоретичних знань і практичних навичок проектування архітектури та розробки крос-платформного програмного забезпечення. Особливий акцент робиться на освоєнні сучасних технологій (React Native, Flutter, Xamarin/.NET MAUI), використанні мов програмування (C#, Dart, JavaScript), створенні адаптивних інтерфейсів, роботі з

базами даних, тестуванні, публікації застосунків та автоматизації процесів розгортання за допомогою CI/CD.

По завершенню вивчення дисципліни студент повинен набути таких знань та вмінь:

Знати:

1. Основні терміни та поняття, пов'язані з крос-платформним програмуванням і розробкою крос-платформних застосунків.
2. Архітектурні моделі та підходи до проектування крос-платформних рішень.
3. Принципи роботи з популярними крос-платформними фреймворками, такими як React Native, Flutter, Xamarin (.NET MAUI) та Electron.
4. Мови програмування, які використовуються для крос-платформного програмування (C#, Dart, JavaScript).
5. Основи розробки адаптивних інтерфейсів користувача для різних платформ.
6. Принципи роботи з локальними та серверними базами даних у крос-платформних застосунках.
7. Технології та методи тестування крос-платформних застосунків (юніт-тести, інтеграційні тести, UI тести).
8. Підходи до публікації та автоматизації процесу розгортання крос-платформних застосунків за допомогою CI/CD.

Вміти:

1. Проектувати та розробляти крос-платформні програмні застосунки з використанням різних фреймворків і мов програмування.
2. Створювати адаптивні інтерфейси користувача для різних платформ, враховуючи їх специфіку.
3. Використовувати популярні інструменти та середовища розробки (Visual Studio, Android Studio, Xcode) для розробки та тестування крос-платформних застосунків.

4. Реалізовувати роботу з базами даних у крос-платформних застосунках, включаючи синхронізацію даних між пристроями та зберігання в хмарі.

5. Використовувати регулярні вирази для валідації вводу та парсингу даних у мобільних і десктопних застосунках.

6. Проводити тестування крос-платформних застосунків та виправляти помилки за допомогою дебагінгу та оптимізації коду.

7. Публікувати та розгортати застосунки на різних платформах (Google Play, App Store) та автоматизувати процес за допомогою CI/CD.

3. Опис навчальної дисципліни

3.1. Лекційні заняття

№	Тема заняття та короткий зміст	Кількість годин	
		ДФН	ЗФН
1.	Тема 1. Вступ до крос-платформного програмування. Поняття крос-платформності, крос-платформних застосунків. Визначення та переваги крос-платформних рішень у програмуванні. Огляд основних інструментів для розробки крос-платформних застосунків.	2	0,5
2.	Тема 2. Архітектура крос-платформних застосунків. Моделі архітектури для крос-платформних застосунків. Підходи до проектування крос-платформних рішень. Взаємодія з операційними системами (Android, iOS, Windows, macOS).	2	0,5
3.	Тема 3. Огляд технологій для крос-платформного програмування. Огляд основних технологій: React Native, Flutter, Xamarin, Electron. Порівняння можливостей та обмежень кожної технології. Вибір інструментів для конкретного проєкту.	2	0,5
4.	Тема 4. Мови програмування для крос-платформних застосунків. Огляд мов програмування для крос-платформних застосунків: JavaScript, Dart, C#. Використання мов в контексті різних фреймворків та платформ. Основи роботи з бібліотеками та фреймворками для крос-платформних застосунків.	2	1
5.	Тема 5. Розробка інтерфейсів користувача для крос-платформних застосунків. Принципи побудови адаптивних інтерфейсів для різних платформ. Інструменти та фреймворки для створення UI (Flutter, React Native). Крос-платформна взаємодія з користувачем (нативні елементи UI та кастомізовані).	2	0,5
6.	Тема 6. Робота з базами даних у крос-платформних застосунках. Локальні бази даних (SQLite, Realm). Підключення до серверних баз даних через API (RESTful, GraphQL). Синхронізація даних між пристроями та зберігання в хмарі.	2	0,5

7.	Тема 7. Регулярні вирази в крос-платформному програмуванні. Поняття регулярних виразів. Основні конструкції регулярних виразів (метасимволи, кваліфікатори, групи, заміни). Використання регулярних виразів у мовах програмування (JavaScript, Dart, C#) та їх застосування в крос-платформних застосунках. Валідація вводу (електронні адреси, телефонні номери тощо) за допомогою регулярних виразів. Застосування регулярних виразів для парсингу даних та пошуку.	2	1
8	Тема 8. Тестування та налагодження крос-платформних застосунків. Підходи до тестування для різних платформ (мобільних та десктопних). Автоматизовані інструменти тестування (Jest, Flutter Test). Важливість юніт-тестів, інтеграційних тестів та UI тестування.	2	1
9.	Тема 9. Публікація та підтримка крос-платформних застосунків. Процес публікації застосунка в магазині Google Play та Apple App Store. Розгортання крос-платформних застосунків на різних пристроях та операційних системах. Використання CI/CD для автоматизації розгортання та оновлень.	2	0,5
Усього годин		18	6

3.2. Лабораторні заняття

№	Тема заняття	Кількість годин	
		ДФН	ЗФН
1.	Створення простого крос-платформного застосунку за допомогою React Native.	6	2
2.	Розробка крос-платформного застосунку за допомогою Flutter.	6	2
3.	Робота з локальними базами даних у крос-платформних застосунках.	6	1
4.	Використання регулярних виразів для валідації вводу.	6	1
5.	Тестування та налагодження крос-платформних застосунків.	6	-
6.	Публікація крос-платформного застосунку.	6	-
Усього годин		36	6

3.3. Самостійна робота

№	Найменування робіт	Кількість годин	
		ДФН	ЗФН
1.	Тема 1. Вступ до крос-платформного програмування. Сучасні інструменти для розробки крос-платформних застосунків. Підготовка до лабораторної роботи 1.	8	12
2.	Опрацювання лекційного матеріалу теми 1. Тема 2. Архітектура крос-платформних застосунків. Взаємодія різних крос-платформних архітектурних рішень з операційними системами (Android, iOS, macOS). Підготовка до лабораторної роботи 1, 2.	8	12
3.	Опрацювання лекційного матеріалу теми 2. Тема 3. Огляд технологій для крос-платформного програмування. Аналіз характеристик та вимог для проєктів із використанням технологій: React Native, Flutter, Xamarin, Electron. Підготовка до лабораторної роботи 1, 2.	9	12
4.	Опрацювання лекційного матеріалу теми 3. Тема 4. Мови програмування для крос-платформних застосунків. Проектування бібліотек для крос-платформних застосунків на мовах JavaScript, Dart, C#, його переваги та недоліки. Підготовка до лабораторної роботи 2.	8	12
5.	Опрацювання лекційного матеріалу теми 4. Тема 5. Розробка інтерфейсів користувача для крос-платформних застосунків. Ознайомлення із нативними та кастомізованими елементами UI при реалізації крос-платформної взаємодії з користувачем. Підготовка до лабораторної роботи 3.	8	12
6.	Опрацювання лекційного матеріалу теми 5. Тема 6. Робота з базами даних у крос-платформних застосунках. Програмна реалізація синхронізації даних між пристроями та використання хмарних сервісів для їх зберігання. Підготовка до лабораторної роботи 3.	8	12
7.	Підготовка до модуля 1 у вигляді тестування.	10	14

8.	Опрацювання лекційного матеріалу теми 6. Тема 7. Регулярні вирази в крос-платформному програмуванні. Програмна реалізація крос-платформного рішення із застосуванням регулярних виразів для парсингу даних та пошуку. Підготовка до лабораторної роботи 4.	9	12
9.	Опрацювання лекційного матеріалу теми 7. Тема 8. Тестування та налагодження крос-платформних застосунків. Порівняльна характеристика підходів до тестування. Вибір одного підходу для тестування крос-платформних рішень, реалізованих у ході виконання лабораторних робіт. Підготовка до лабораторної роботи 5.	9	14
10.	Опрацювання лекційного матеріалу тем 8. Тема 9. Публікація та підтримка крос-платформних застосунків. Автоматизація процесу розгортання та оновлення крос-платформних застосунків із використанням CI/CD. Підготовка до лабораторної роботи 6.	9	12
11.	Підготовка до модуля 2 у вигляді тестування.	10	14
Усього годин		96	138

4. Критерії оцінювання результатів навчання студентів

Форма підсумкового семестрового контролю – залік

Підсумкова семестрова оцінка заліку складається з суми балів, отриманих студентом при проведенні проміжних (модульних) контролів рівня засвоєння теоретичних знань (за перший та другий модулі) та отриманих балів за лабораторні роботи.

Захист звіту з лабораторної роботи оцінюється відповідною кількістю балів поданою в таблиці. Після проходження теоретичного матеріалу проводиться електронне тестування його засвоєння у вигляді проміжного (модульного) контролю.

Контроль здійснюється засобами електронного навчального курсу (ЕНК) на сервері дистанційного навчання <http://dl.tntu.edu.ua>, ідентифікатор дисципліни ID 6690.

Модуль 1			Модуль 2			Підсумковий контроль	Разом з дисципліни
Аудиторна та самостійна робота			Аудиторна та самостійна робота				
Теоретичний курс (тестування)	Лабораторна робота		Теоретичний курс (тестування)	Лабораторна робота			
17	21		16	21		25	100
№ тем	Вид робіт	бал	№ тем	Вид робіт	бал	за кожних три бали семестрової оцінки студент отримує 1 бал підсумкової семестрової оцінки автоматично	
Тема 1	Лаб. роб. №1	7	Тема 7	Лаб. роб. № 4	7		
Тема 2							
Тема 3	Лаб. роб. №2	7	Тема 8	Лаб. роб. №5	7		
Тема 4			Тема 9	Лаб. роб. №6	7		
Тема 5							
Тема 6	Лаб. роб. №3	7					

До підсумкового семестрового контролю (складання семестрового заліку) допускаються студенти, які протягом семестру виконали всі види навчальної роботи, успішно пройшли проміжні (модульні) контролі і набрали не менше 45 балів семестрової бальної оцінки та за умови отримання не менше 60% балів за результатами кожного проміжного (модульного) контролю рівня знань.

Підсумкова оцінка записується за 100-бальною шкалою із подальшим переведенням її у шкалу Європейської кредитно-трансферної системи (ECTS) відповідно А, В, С, D, E, F, FX при цьому чотирибальна шкала оцінок (з записом семестрової оцінки «відмінно» - А, «добре» - В,С, «задовільно» - D, E відповідають підсумковому результату «зараховано», «незадовільно» - F, FX відповідає підсумковому результату «незараховано»).

5. Навчально-методичне забезпечення

1. Методичні вказівки для самостійної роботи студентів з дисципліни «Крос-платформне програмування» для спеціальності 122 «Комп'ютерні науки».

2. Методичні вказівки для виконання лабораторних робіт з дисципліни «Крос-платформне програмування» для студентів спеціальності 122 «Комп'ютерні науки».

3. Лекції в електронному навчальному курсі «Крос-платформне програмування» для студентів напряму підготовки 122 «Комп'ютерні науки», ідентифікаційний номер якого ID 6690, <https://dl.tntu.edu.ua/>

6. Рекомендована література

Базова

1. Clean Architecture: A Craftsman's Guide to Software Structure and Design (Robert C. Martin Series) ISBN-13: 9780134494166.
2. Farooq, M. S., Riaz, S., Alvi, A., Ali, A., & Rehman, I. U. (2022). Cross-Platform Mobile Development Approaches and Frameworks. VFAST Transactions on Software Engineering, 10(2), 79–93. <https://doi.org/10.21015/vtse.v10i2.978>
3. Ameen, S. Y. ., & Mohammed, D. Y. (2022). Developing Cross-Platform Library Using Flutter. European Journal of Engineering and Technology Research, 7(2), 18–21. <https://doi.org/10.24018/ejeng.2022.7.2.2740>
4. Tollin, G., & Marcus, L. (2023). React Native vs. Flutter : A performance comparison between cross-platform mobile application development frameworks. DIVA. [Електронний ресурс]. – Режим доступу: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1768521&dswid=-8495>
5. Systä, K., & Sand, A. (2023). EXPERIMENTS COMPARING CROSS-PLATFORM AND NATIVE MOBILE APPLICATIONS A Systematic Literature Review. [Електронний ресурс]. – Режим доступу: <https://trepo.tuni.fi/bitstream/handle/10024/148166/BarnettAndrew.pdf?sequence=2>

Допоміжна

1. Colombo, C., & Pace, G. J. (2022). Regular Expressions. Runtime Verification, 87–108. https://doi.org/10.1007/978-3-031-09268-8_8
2. Park, J., An, S., Youn, D., Kim, G., & Ryu, S. (2021). JEST: N+1-Version Differential Testing of Both JavaScript Engines and Specification. ArXiv (Cornell University). <https://doi.org/10.1109/icse43902.2021.00015>
3. Beyer, D. (2023). Software Testing: 5th Comparative Evaluation: Test-Comp 2023. Lecture Notes in Computer Science, 309–323. https://doi.org/10.1007/978-3-031-30826-0_17

7. Інформаційні ресурси

1. Електронний навчальний курс «Крос-платформне програмування» в системі дистанційного навчання ТНТУ імені І. Пулюя. [Електронний ресурс]. – Режим доступу: <https://dl.tntu.edu.ua/bounce.php?course=6690>
2. Introduction · React Native. [Електронний ресурс]. – Режим доступу: <https://reactnative.dev/docs/getting-started>
3. Flutter documentation. [Електронний ресурс]. – Режим доступу: <https://docs.flutter.dev/>
4. Xamarin documentation. [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/uk-UA/xamarin>
5. Youtube-канал про розробку крос-платформних застосунків з використанням Flutter. [Електронний ресурс]. – Режим доступу: <https://www.youtube.com/@flutterdev>
6. Youtube-канал про розробку крос-платформних застосунків з використанням React Native. [Електронний ресурс]. – Режим доступу: https://www.youtube.com/watch?v=0-S5a0eXPoc&ab_channel=ProgrammingwithMosh

8. Зміни та доповнення до робочої програми навчальної дисципліни

№	Зміст внесених змін (доповнень)	Дата і № протоколу засідання кафедри	Примітки