

Чернівецький національний університет імені Юрія Федьковича
Навчально-науковий інститут фізико-технічних та комп'ютерних наук
Кафедра комп'ютерних наук

«ЗАТВЕРДЖУЮ»
директор навчально-наукового
інституту фізико-технічних та
комп'ютерних наук, к.ф.-м.н., доцент
_____ Петро ШПАТАР
«___» _____ 2025 р.

РОБОЧА ПРОГРАМА
навчальної дисципліни
ТЕХНОЛОГІЇ .NET РОЗРОБКИ МОБІЛЬНИХ АПЛІКАЦІЙ
(назва навчальної дисципліни)

обов'язкова
(вказати: обов'язкова / вибіркова)

Освітньо-професійна програма **«Інтелектуальний аналіз даних в комп'ютерних інформаційних системах»**
(назва програми)

Спеціальність **ФЗ «Комп'ютерні науки»**
(вказати: код, назва)

Галузь знань **Е «Інформаційні технології»**
(вказати: шифр, назва)

Рівень вищої освіти: **перший (бакалаврський)**
(вказати: перший бакалаврський / другий магістерський)

навчально-науковий інститут фізико-технічних та комп'ютерних наук
(назва факультету / навчально-наукового інституту,
на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)

мова навчання: **українська**
(вказати, на якій мові читається навчальна дисципліна)

Чернівці 2025 рік

Робоча програма навчальної дисципліни «Технології .NET розробки мобільних аплікацій» **складена відповідно до** освітньо-професійної програми «Інтелектуальний аналіз даних в комп'ютерних інформаційних системах» спеціальності F3 Комп'ютерні науки галузі знань F Інформаційні технології, затвердженої вченою радою Чернівецького національного університету імені Юрія Федьковича (Протокол №6 від «26» травня 2025 року)

Розробник: Томка Ю.Я., доцент кафедри комп'ютерних наук, к.ф.-м.н., доцент.

Викладач: Томка Ю.Я., доцент кафедри комп'ютерних наук, к.ф.-м.н., доцент.

Погоджено з гарантом ОП _____ Дмитро УГРИН

Затверджено на засіданні кафедри комп'ютерних наук

Протокол № 1 від “27” серпня 2025 року

Завідувач кафедри

Юрій УШЕНКО

Схвалено методичною радою навчально-наукового інституту фізико-технічних та комп'ютерних наук

Протокол № 1 від “28” серпня 2025 року

Голова методичної ради _____ Іван КОЗЯРСЬКИЙ

Мета навчальної дисципліни. Метою викладання дисципліни «Технології .NET розробки мобільних аплікацій» є підготовка кваліфікованих спеціалістів, здатних проектувати, розробляти та впроваджувати сучасні крос-платформні мобільні рішення на основі платформи .NET MAUI (Multi-platform App UI) та мови програмування C#. Дисципліна спрямована на формування комплексних компетентностей у галузі мобільної розробки, включаючи створення користувацьких інтерфейсів, реалізацію бізнес-логіки, інтеграцію з хмарними сервісами та забезпечення високої продуктивності застосунків для платформ iOS, Android та Windows.

Пререквізити. Навчальна дисципліна «Технології .NET розробки мобільних аплікацій» ґрунтується на дисципліні «Алгоритмізація та програмування», «Об'єктно-орієнтоване програмування» та є основою для дисципліни «Технології .NET у сучасній веб-інженерії», «Технології створення програмних продуктів».

Результати навчання:

Під час вивчення даної дисципліни студенти набудуть:

- I. загальних компетентностей:
 - **ЗК1.**Здатність до абстрактного мислення, аналізу та синтезу.
 - **ЗК2.**Здатність застосовувати знання у практичних ситуаціях.
 - **ЗК3.**Знання та розуміння предметної області та розуміння професійної діяльності.
 - **ЗК4.**Здатність спілкуватися державною мовою як усно, так і письмово.
 - **ЗК5.**Здатність спілкуватися іноземною мовою.
 - **ЗК6.**Здатність вчитися і оволодівати сучасними знаннями.
 - **ЗК7.**Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
 - **ЗК8.**Здатність генерувати нові ідеї (креативність).
 - **ЗК10.**Здатність бути критичним і самокритичним.
 - **ЗК11.**Здатність приймати обґрунтовані рішення.
 - **ЗК12.**Здатність оцінювати та забезпечувати якість виконуваних робіт.
- II. спеціальних (фахових, предметних) компетентностей:
 - **СК08.** Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальнення, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами та алгоритмами обчислень, структурами даних і механізмами управління.
 - **СК12.** Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення.
 - **СК14.** Здатність застосовувати методи та засоби забезпечення інформаційної безпеки, розробляти й експлуатувати спеціальне програмне забезпечення захисту інформаційних ресурсів об'єктів критичної інформаційної інфраструктури.
- III. Програмними результатами навчання є:
 - **ПР05.** Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.
 - **ПР09.** 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 застосунків, 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.

- **ПР15.** Розуміти концепцію інформаційної безпеки, принципи безпечного проектування програмного забезпечення, забезпечувати безпеку комп'ютерних мереж в умовах неповноти та невизначеності вихідних даних.

Опис навчальної дисципліни Загальна інформація

Назва навчальної дисципліни «Технології .NET розробки мобільних аплікацій»											
Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	2	3	3	90	14			16	60		залік

Структура змісту навчальної дисципліни

Назви змістових модулів і тем навчальних занять	Кількість годин					
	денна форма					
	усього	у тому числі				
		лекц	практ.	лаб.	інд.	срс
1	2	3	4	5	6	7
Змістовий модуль 1. ОСНОВИ РОЗРОБКИ ТА АРХІТЕКТУРНІ КОНЦЕПЦІЇ МОБІЛЬНИХ АПЛІКАЦІЙ						
Тема 1. Вступ до .NET розробки мобільних застосунків	7	1		-		6
Тема 2. Створення інтерфейсу користувача (UI/UX)	10	2		2		6
Тема 3. Архітектурний патерн MVVM	9	1		2		6
Тема 4. Розширені техніки роботи з XAML та UI	10	2		2		6
Тема 5. Робота з базами даних та CRUD-операції на основі Entity Framework	10	1		2		6
Разом за ЗМ 1	46	7	0	8	0	30
Змістовий модуль 2. РОЗШИРЕНІ МОЖЛИВОСТІ ПЛАТФОРМИ						
Тема 6. Аутентифікація та авторизація користувачів	9	2		2		6
Тема 7. Практичні сценарії: AI, SignalR та реал-тайм комунікації	7	2		2		6
Тема 8. Платформно-специфічні API та власні обробники	7	1		2		6
Тема 9. Оптимізація продуктивності застосунку	7	1		-		6
Тема 10. Основи тестування та налагодження мобільних застосунків	9	1		2		6
Разом за ЗМ 2	61	7	0	8	0	30

Усього годин	120	14	0	16	0	60
--------------	-----	----	---	----	---	----

Тематика лекційних занять з переліком питань

№	Назва теми з основними питаннями
1.	Тема 1. Вступ до .NET розробки мобільних застосунків План лекції: 1. Огляд екосистеми .NET та еволюція мобільної розробки (Xamarin → .NET MAUI) 2. Архітектура .NET MAUI: принципи крос-платформної розробки 3. Налаштування середовища розробки (Visual Studio, SDK, емулятори) 4. Структура проєкту .NET MAUI: файли, папки, конфігурація 5. Життєвий цикл мобільного застосунку 6. Перший мобільний застосунок: Hello World 7. Основи компіляції та розгортання на різних платформах (iOS, Android, Windows) 8. Інструменти налагодження та тестування
2.	Тема 2. Створення інтерфейсу користувача (UI/UX) План лекції: 1. Основи XAML: синтаксис, структура, принципи декларативного опису UI 2. Контейнери компоновання: StackLayout, Grid, FlexLayout, AbsoluteLayout 3. Основні елементи керування: Label, Entry, Button, Image, Picker 4. Робота з ресурсами та стилями: ResourceDictionary, Style, StaticResource 5. Адаптивний дизайн: підтримка різних розмірів екранів та орієнтацій 6. Принципи Material Design та Human Interface Guidelines 7. Колірні схеми, шрифти та візуальна ієрархія 8. Доступність (Accessibility) у мобільних застосунках
3.	Тема 3. Архітектурний патерн MVVM План лекції: 1. Архітектурні патерни у мобільній розробці: MVC, MVP, MVVM 2. Принципи MVVM: Model, View, ViewModel та їх взаємодія 3. Data Binding: односпрямований та двоспрямований зв'язок даних 4. INotifyPropertyChanged: реалізація реактивності 5. Commands: ICommand та RelayCommand для обробки подій 6. Dependency Injection у .NET MAUI 7. Організація коду та структура проєкту згідно з MVVM 8. Бібліотеки та фреймворки: CommunityToolkit.Mvvm, Prism
4.	Тема 4. Розширені техніки роботи з XAML та UI План лекції: 1. Створення власних елементів керування (Custom Controls) 2. Шаблони даних: DataTemplate та ControlTemplate 3. Колекції та списки: CollectionView, ListView, CarouselView 4. Навігація між сторінками: Shell, NavigationPage, TabbedPage 5. Анімації та переходи: Animation API у .NET MAUI 6. Жести та взаємодія: TapGesture, SwipeGesture, PinchGesture 7. Графіка та малювання: Microsoft.Maui.Graphics 8. Робота з медіа: відео, аудіо, камера
5.	Тема 5. Робота з базами даних та CRUD-операції на основі Entity Framework План лекції: 1. Типи баз даних для мобільних застосунків: SQLite, локальне сховище 2. Встановлення та налаштування Entity Framework Core 3. Моделі даних: створення класів-сутностей та DbContext 4. Міграції: створення та застосування змін у структурі БД 5. CRUD-операції: Create, Read, Update, Delete 6. Асинхронна робота з даними: async/await 7. Відносини між сутностями: One-to-One, One-to-Many, Many-to-Many

	8. Репозиторій та Unit of Work: паттерни доступу до даних
6.	Тема 6. Аутентифікація та авторизація користувачів План лекції: 1. Основи безпеки у мобільних застосунках 2. Типи аутентифікації: форми, OAuth 2.0, OpenID Connect 3. Реалізація системи реєстрації та входу користувачів 4. Безпечне зберігання облікових даних: SecureStorage API 5. Токени доступу: JWT (JSON Web Tokens) 6. Інтеграція з зовнішніми провайдерами: Google, Facebook, Microsoft 7. Управління сесіями та автоматичне оновлення токенів 8. Авторизація: ролі, права доступу, захист ресурсів
7.	Тема 7. Практичні сценарії: AI, SignalR та реал-тайм комунікації План лекції: 1. Інтеграція штучного інтелекту у мобільні застосунки 2. Використання Azure Cognitive Services: Computer Vision, Text Analytics 3. ML.NET: машинне навчання на пристрої 4. Основи SignalR: архітектура клієнт-серверної взаємодії 5. Налаштування SignalR Hub та підключення клієнта 6. Реал-тайм чат: відправлення та отримання повідомлень 7. Сповіщення та оновлення у реальному часі 8. Обробка помилок та відновлення з'єднання
8.	Тема 8. Платформо-специфічні API та власні обробники План лекції: 1. Доступ до нативних можливостей платформ через .NET MAUI 2. Essentials API: геолокація, сенсори, файлова система 3. Створення Platform Handlers для кастомізації елементів 4. Умовна компіляція: директиви препроцесора для різних платформ 5. Виклик нативного коду: Platform Invocation Services (P/Invoke) 6. Робота з нативними бібліотеками: Bindings 7. Дозволи (Permissions): запит та обробка 8. Push-нотифікації: Firebase Cloud Messaging, Apple Push Notification Service
9.	Тема 9. Оптимізація продуктивності застосунку План лекції: 1. Профілювання та аналіз продуктивності мобільних застосунків 2. Оптимізація використання пам'яті: виявлення витоків (memory leaks) 3. Ліниве завантаження (Lazy Loading) та віртуалізація списків 4. Кешування даних: стратегії та реалізація 5. Оптимізація мережових запитів: compression, batching 6. Асинхронність та багатопотоковість: Task, Thread, ThreadPool 7. Оптимізація зображень: стиснення, формати, завантаження 8. Моніторинг продуктивності у production: Application Insights
10.	Тема 10. Основи тестування та налагодження мобільних застосунків План лекції: 1. Типи тестування: Unit, Integration, UI, E2E 2. Фреймворки для тестування: xUnit, NUnit, MSTest 3. Unit-тестування: тестування бізнес-логіки та ViewModel 4. Mocking та Stubbing: Moq, FakeItEasy 5. UI-тестування: Appium, Xamarin.UITest 6. Налаштування: breakpoints, watch, call stack 7. Логування: ILogger, Serilog, Application Insights 8. Continuous Integration/Continuous Deployment (CI/CD): GitHub Actions, Azure DevOps

Тематика лабораторних занять з переліком питань

№	Назва теми (завдання)
1.	Лабораторна робота №1. Основи створення крос-платформних мобільних апікацій у .NET MAUI
2.	Лабораторна робота №2. Розробка інтерфейсу користувача засобами XAML
3.	Лабораторна робота №3. Кастомізація макетів сторінок у .NET MAUI XAML
4.	Лабораторна робота №4. Імплементация механізму Data Binding у .NET MAUI
5.	Лабораторна робота №5. Реалізація архітектурного патерну MVVM
6.	Лабораторна робота №6. Використання спільних ресурсів для консистентного дизайну
7.	Лабораторна робота №7. Створення багатосторінкової апікації з Tab та Flyout навігацією
8.	Лабораторна робота №8. Інтеграція локального сховища даних SQLite

Індивідуальні науково-дослідні завдання (ІНДЗ)

№	Завдання до тем
1.	Порівняльний аналіз крос-платформних фреймворків для мобільної розробки <i>Дослідити:</i> .NET MAUI, Flutter, React Native, Xamarin.Forms <i>Створити:</i> тестову апікацію на 2-3 фреймворках з ідентичним функціоналом, провести бенчмарк-тести, підготувати звіт з порівняльними метриками
2.	Оптимізація продуктивності CollectionView у .NET MAUI <i>Дослідити:</i> стратегії віртуалізації, кешування, data templates <i>Створити:</i> демо-апікацію з великими списками (10000+ елементів), реалізувати різні підходи оптимізації, задокументувати результати профілювання
3.	Реалізація офлайн-режиму у мобільних апікаціях <i>Дослідити:</i> синхронізація даних, conflict resolution, offline-first архітектура <i>Створити:</i> апікацію з можливістю роботи без інтернету, автоматичною синхронізацією при відновленні з'єднання, обробкою конфліктів
4.	Патерни навігації у MVVM апікаціях <i>Дослідити:</i> Shell Navigation, NavigationService, параметри навігації <i>Створити:</i> прототип апікації з реалізацією різних типів навігації, порівняти підходи, написати рекомендації
5.	Безпека мобільних апікацій: best practices <i>Дослідити:</i> шифрування даних, SecureStorage, certificate pinning, code obfuscation <i>Створити:</i> демо-апікацію з реалізацією різних механізмів безпеки, провести тестування на вразливості, підготувати security checklist
6.	Інтеграція машинного навчання у .NET MAUI апікації <i>Дослідити:</i> ML.NET, ONNX Runtime, Azure Cognitive Services <i>Створити:</i> апікацію з розпізнаванням зображень або текстів, порівняти on-device та cloud підходи, виміряти швидкість та точність
7.	Архітектурні патерни для enterprise мобільних рішень <i>Дослідити:</i> Clean Architecture, Onion Architecture, Hexagonal Architecture <i>Створити:</i> референсну апікацію з багатошаровою архітектурою, dependency injection, репозиторіями, задокументувати структуру
8.	Робота з нативними модулями через Platform Handlers <i>Дослідити:</i> створення custom handlers, маппінг властивостей, platform-specific API <i>Створити:</i> кастомний контрол з нативною реалізацією для iOS та Android, написати документацію з використання

9.	Стратегії тестування мобільних застосунків <i>Дослідити:</i> unit testing, integration testing, UI testing, test automation <i>Створити:</i> застосунок з комплексним покриттям тестами (мінімум 70%), налаштувати test runner, підготувати звіт про покриття
10.	Continuous Integration/Continuous Deployment для .NET MAUI <i>Дослідити:</i> GitHub Actions, Azure DevOps, App Center <i>Створити:</i> повний CI/CD pipeline для застосунку з автоматичним білдом, тестуванням та деплоєм на TestFlight/Firebase
11.	Accessibility у мобільних застосунках <i>Дослідити:</i> SemanticProperties, screen readers, контрастність, розміри шрифтів <i>Створити:</i> застосунок з повною accessibility підтримкою, протестувати з VoiceOver/TalkBack, підготувати accessibility audit
12.	Управління станом у складних .NET MAUI застосунках <i>Дослідити:</i> Redux-подібні патерни, Reactive Extensions, state management libraries <i>Створити:</i> застосунок з централізованим state management, реалізувати actions/reducers, порівняти з традиційним MVVM
13.	Push-нотифікації: архітектура та реалізація <i>Дослідити:</i> FCM, APNS, Azure Notification Hubs, локальні нотифікації <i>Створити:</i> застосунок з push-нотифікаціями, обробкою foreground/background, deep linking, аналітикою відкриттів
14.	Локалізація та інтернаціоналізація мобільних застосунків <i>Дослідити:</i> resource files, культури, RTL підтримка, динамічна зміна мови <i>Створити:</i> багатомовну застосунок (мінімум 3 мови), реалізувати зміну мови без перезапуску, підтримати RTL для арабської
15.	Робота з камерою та медіа-файлами <i>Дослідити:</i> Camera API, MediaPicker, обробка зображень, компресія відео <i>Створити:</i> застосунок для роботи з фото/відео, реалізувати фільтри, компресію, збереження в галерею
16.	Геолокація та картографічні сервіси <i>Дослідити:</i> Geolocation API, Google Maps, background location tracking <i>Створити:</i> застосунок з відстеженням маршруту, геофенсингом, відображенням на карті, збереженням треків
17.	Міграція з Xamarin.Forms на .NET MAUI <i>Дослідити:</i> breaking changes, migration toolkit, compatibility issues <i>Створити:</i> взяти існуючий Xamarin.Forms проєкт, мігрувати на .NET MAUI, задокументувати всі проблеми та рішення
18.	Анімації та мікроінтеракції у мобільному UI <i>Дослідити:</i> Animation API, Lottie, custom animations, performance impact <i>Створити:</i> застосунок-демо з різними типами анімацій, мікроінтеракціями, smooth transitions, виміряти FPS
19.	Робота з великими даними у мобільних застосунках <i>Дослідити:</i> pagination, lazy loading, data virtualization, SQLite optimization <i>Створити:</i> застосунок з базою 50000+ записів, реалізувати ефективний пошук, фільтрацію, сортування, виміряти продуктивність
20.	Монетизація мобільних застосунків <i>Дослідити:</i> in-app purchases, subscriptions, ads integration, payment gateways <i>Створити:</i> застосунок з інтеграцією платежів (Stripe/PayPal), підписками, рекламою (AdMob), підготувати документацію з налаштування

Завдання для самостійної роботи студентів

№	Назва теми	Завдання для самостійної роботи
1	Тема 1. Вступ до .NET розробки мобільних застосунків	1. Опрацювати історію розвитку крос-платформних технологій від Xamarin до .NET MAUI 2. Дослідити архітектуру .NET MAUI та принципи крос-платформної розробки. 3. Скласти порівняльну таблицю .NET MAUI, Flutter, React Native 4. Налаштувати робоче середовище (Visual Studio, SDK, емулятори) та створити перший Hello World застосунок.
2	Тема 2. Створення	1. Ознайомитися з основами синтаксису XAML та його

	інтерфейсу користувача (UI/UX)	структурою 2. Вивчити різні типи контейнерів компонування (StackLayout, Grid, FlexLayout). 3. Створити декілька варіантів макетів однієї сторінки з різними layouts. 4. Проаналізувати принципи Material Design та Human Interface Guidelines 5. Розробити адаптивний інтерфейс для різних розмірів екранів.
3	Тема 3. Архітектурний патерн MVVM	1. Вивчити теоретичні основи патернів MVC, MVP, MVVM та їх відмінності 2. Дослідити принципи Data Binding та механізм INotifyPropertyChanged 3. Створити просту аплікацію з реалізацією MVVM без використання фреймворків 4. Опрацювати документацію CommunityToolkit.Mvvm 5. Реалізувати Commands для обробки подій у ViewModel.
4	Тема 4. Розширені техніки роботи з XAML та UI	1. Вивчити принципи створення Custom Controls та User Controls 2. Опрацювати механізми DataTemplate та ControlTemplate 3. Створити аплікацію з CollectionView та кастомними шаблонами елементів 4. Реалізувати навігацію між сторінками з використанням Shell 5. Додати анімації та жести до інтерфейсу. 6. Експериментувати з Microsoft.Maui.Graphics для малювання.
5	Тема 5. Робота з базами даних та CRUD-операції на основі Entity Framework	1. Встановити та налаштувати Entity Framework Core для мобільної аплікації 2. Створити моделі даних та DbContext для локальної SQLite бази 3. Реалізувати міграції та застосувати їх до бази даних 4. Розробити повний цикл CRUD-операцій (Create, Read, Update, Delete) 5. Впровадити асинхронну роботу з даними через async/await. 6. Реалізувати Repository pattern для інкапсуляції логіки доступу до даних.
6	Тема 6. Аутентифікація та авторизація користувачів	1. Вивчити основи протоколів OAuth 2.0 та OpenID Connect 2. Дослідити механізми безпечного зберігання даних через SecureStorage API 3. Реалізувати систему реєстрації та входу користувачів 4. Опрацювати структуру JWT токенів та їх використання 5. Створити демо-аплікацію з інтеграцією Social Login (Google або Microsoft) 6. Впровадити механізм оновлення токенів (Refresh Token).
7	Тема 7. Практичні сценарії: AI, SignalR та реал-тайм комунікації	1. Ознайомитися з можливостями Azure Cognitive Services 2. Вивчити основи SignalR та архітектуру Hub 3. Створити простий чат-застосунок з використанням SignalR 4. Дослідити ML.NET та можливості машинного навчання на пристрої 5. Реалізувати обробку помилок та відновлення з'єднання у реал-тайм комунікаціях.
8	Тема 8. Платформоспецифічні API та власні обробники	1. Опрацювати документацію .NET MAUI Essentials API 2. Вивчити принципи створення Platform Handlers 3. Реалізувати доступ до камери, геолокації та сенсорів пристрою

		4. Створити Custom Handler для платформи-специфічного контролю 5. Дослідити систему дозволів (Permissions) на Android та iOS 6. Впровадити Push-нотифікації через Firebase Cloud Messaging.
9	Тема 9. Оптимізація продуктивності застосунку	1. Ознайомитися з інструментами профілювання у Visual Studio 2. Дослідити типові причини витоків пам'яті та методи їх виявлення 3. Реалізувати lazy loading та віртуалізацію для великих списків 4. Впровадити кешування даних (in-memory та disk cache) 5. Оптимізувати завантаження та відображення зображень 6. Провести бенчмарк-тестування до та після оптимізації 7. Налаштувати моніторинг продуктивності через Application Insights.
10	Тема 10. Основи тестування та налагодження мобільних аплікацій	1. Вивчити принципи Test-Driven Development (TDD) 2. Опрацювати фреймворки xUnit, NUnit для створення тестів 3. Написати unit-тести для ViewModel та бізнес-логіки 4. Дослідити мокування залежностей через Moq або FakeItEasy 5. Налаштувати CI/CD pipeline у GitHub Actions або Azure DevOps 6. Створити integration тести для роботи з базою даних 7. Впровадити логування через Serilog з різними рівнями

Методи навчання

У процесі викладання дисципліни «Технології .NET розробки мобільних аплікацій» застосовуються сучасні методи навчання, спрямовані на формування у студентів теоретичних знань і практичних навичок створення крос-платформних мобільних рішень, розвитку алгоритмічного мислення та здатності до вирішення складних інженерних завдань.

Основні методи навчання:

1. *Пояснювально-ілюстративні методи* – використання мультимедійних презентацій, демонстрація архітектури .NET MAUI, аналіз коду C# та XAML, огляд сучасних практик мобільної розробки та UI/UX паттернів.
2. *Репродуктивні методи* – виконання студентами типових завдань із створення користувацьких інтерфейсів, реалізації CRUD-операцій, застосування патерну MVVM, налаштування навігації та зв'язування даних.
3. *Проблемно-пошукові методи* – постановка проблемних завдань, що вимагають самостійного пошуку оптимальних архітектурних рішень, вибору відповідних патернів проєктування, оптимізації продуктивності та вирішення платформи-специфічних проблем.
4. *Практичні методи* – створення навчальних мобільних застосунків, розробка прототипів, використання інструментів розробки (Visual Studio, Android Studio, Xcode), налагодження на емуляторах і фізичних пристроях, робота з системами контролю версій (Git).
5. *Інтерактивні методи навчання* – робота в малих групах над спільними проєктами, code review, колективне обговорення архітектурних рішень, peer programming, взаємооцінювання студентських застосунків.
6. *Демонстраційні методи* – показ покрокового процесу створення мобільного застосунку, live coding сесії, демонстрація інтеграції з API та базами даних, використання інструментів налагодження та профілювання продуктивності.
7. *Проектно-дослідницький метод* – виконання індивідуальних науково-дослідних завдань (ІНДЗ) і підсумкового мобільного проєкту з повним циклом розробки: від проєктування архітектури та створення UI до тестування, оптимізації та підготовки до публікації в магазинах додатків.

8. *Методи самостійної роботи* – опрацювання технічної документації Microsoft, аналіз open-source проєктів на GitHub, вивчення best practices мобільної розробки, участь у онлайн-курсах та вебінарах, створення власного портфоліо мобільних застосунків.
9. *Кейс-метод* – аналіз реальних сценаріїв розробки корпоративних мобільних рішень, розбір типових помилок та антипатернів, вивчення успішних кейсів впровадження .NET MAUI у production середовищі.
10. *Agile-методології* – застосування Scrum/Kanban підходів до організації навчальних проєктів, проведення sprint planning, daily stand-ups, retrospectives для формування навичок командної розробки.

Система контролю та оцінювання

Оцінювання результатів навчальної діяльності студентів з дисципліни «Технології .NET розробки мобільних аплікацій» здійснюється відповідно до вимог кредитно-модульної системи організації навчального процесу та базується на принципах об'єктивності, прозорості, індивідуальності та систематичності контролю.

Методи поточного контролю:

- *Експрес-опитування* – короткі усні або письмові запитання для перевірки розуміння теоретичного матеріалу лекції, основних концепцій .NET MAUI, принципів MVVM, особливостей крос-платформної розробки.
- *Індивідуальне опитування* – усна або письмова форма контролю знань окремих студентів щодо архітектурних патернів, життєвого циклу аплікацій, механізмів зв'язування даних, роботи з базами даних, особливостей платформи-специфічних API.
- *Фронтальне опитування* – колективне обговорення тем, концепцій, аналіз коду C# та XAML, порівняння різних підходів до розробки під час лекцій або лабораторних занять.
- *Оцінювання лабораторних робіт* – перевірка правильності виконання практичних завдань із створення користувацьких інтерфейсів, реалізації MVVM архітектури, інтеграції з базами даних через Entity Framework, впровадження аутентифікації, оптимізації продуктивності аплікацій.
- *Тестування* – електронні або письмові тести для перевірки теоретичних знань синтаксису C#, принципів XAML, архітектурних патернів, особливостей .NET MAUI API, методів роботи з даними, механізмів безпеки.
- *Code Review* – аналіз якості написаного студентами коду з точки зору дотримання стандартів, читабельності, ефективності, використання best practices, архітектурної чистоти.
- *Мініпрактикуми* – короткі практичні завдання з реалізації окремих компонентів (custom controls, data templates, navigation flows, API integration), що виконуються на занятті з подальшим обговоренням рішень.
- *Захист лабораторних робіт* – демонстрація функціональності створеної аплікації на емуляторі або фізичному пристрої, пояснення архітектурних рішень, відповіді на запитання викладача щодо реалізації.
- *Захист індивідуального науково-дослідного завдання (ІНДЗ)* – представлення студентом результатів індивідуального дослідження або розробки повнофункціональної мобільної аплікації з обґрунтуванням вибору архітектури, технологічного стеку, дизайнерських і технічних рішень, демонстрацією роботи на різних платформах.
- *Самостійна робота студентів* – оцінюється за результатами виконання домашніх завдань, створення навчальних проєктів, участі в code review, опрацювання технічної документації, вивчення додаткових матеріалів, підготовки презентацій.
- *Модульний контроль* – письмове тестування або практичне завдання після завершення кожного змістового модуля для перевірки засвоєння комплексу знань та вмінь з відповідних тем.

Критерії оцінювання поточного та підсумкового контролю

Контроль та оцінювання результатів навчання здійснюється згідно Положення про контроль і систему оцінювання результатів навчання здобувачів вищої освіти у Чернівецькому національному університеті імені Юрія Федьковича (https://www.chnu.edu.ua/media/geupxdun/polozhennia-pro-kontrol-i-systemu-otsiniuvannia_2020.pdf).

Поточний модульний контроль (ПМК) здійснюється під час проведення лекційних, практичних та індивідуально-консультативних занять з метою перевірки рівня засвоєння теоретичних знань та практичних навичок студента.

Поточний контроль проводиться у таких формах:

- усне опитування та експрес-тестування, що дозволяють швидко перевірити базові знання з теоретичних тем (архітектура .NET MAUI, життєвий цикл аплікацій, синтаксис XAML, принципи MVVM, робота з Entity Framework);
- практичні та лабораторні завдання, спрямовані на розроблення мобільних аплікацій із застосуванням C#, XAML, реалізацією патерну MVVM, створенням адаптивних інтерфейсів, інтеграцією з базами даних, впровадженням аутентифікації та платформо-специфічних можливостей;
- індивідуальні проєкти (повнофункціональні мобільні аплікації з CRUD-операціями, навігацією, роботою з API), які відображають уміння студента комплексно застосовувати набуті знання та створювати готові до публікації рішення;
- code review та обговорення архітектурних рішень, презентації власних проєктів, демонстрація роботи аплікацій на різних платформах та участь у колективному аналізі коду;
- тематичні тестування після завершення ключових модулів курсу (основи .NET MAUI, MVVM та UI, робота з даними, розширені можливості платформи).

Оцінювання результатів поточного контролю базується на таких критеріях:

- повнота та правильність виконання завдань, функціональна відповідність вимогам;
- логічність і коректність архітектури аплікації, дотримання принципів MVVM та розділення відповідальностей;
- правильність використання C# конструкцій, XAML-розмітки, API .NET MAUI, механізмів зв'язування даних;
- якість коду: читабельність, дотримання code conventions, використання best practices, відсутність code smells;
- коректність роботи з даними: реалізація CRUD-операцій, обробка помилок, асинхронність;
- функціональність інтерфейсу: зручність навігації, адаптивність під різні розміри екранів, відповідність принципам UX/UI для мобільних платформ;
- продуктивність та оптимізація: ефективне використання ресурсів, відсутність витоків пам'яті, швидкість відгуку інтерфейсу;
- крос-платформна сумісність: коректна робота на iOS, Android та Windows (де застосовно);
- безпека: правильна реалізація аутентифікації, захист даних, використання SecureStorage;
- наявність тестів: покриття unit-тестами критичної бізнес-логіки та ViewModel;
- документація коду: наявність коментарів, XML-документації для публічних методів;
- своєчасність подання роботи, самостійність її виконання, здатність пояснити та обґрунтувати прийняті рішення.

Відвідування занять. Відсутність на аудиторному занятті не передбачає нарахування штрафних балів, оскільки фінальний рейтинговий бал студента формується виключно на основі

оцінювання результатів навчання. Разом з тим, обговорення результатів виконання тематичних завдань, а також презентація / публічний виступ та участь у обговореннях та доповнення на практичних заняттях оцінюватимуться під час аудиторних занять.

Оцінювання пропущених контрольних заходів. Кожен студент має право відпрацювати пропущені з поважної причини (лікарняний, мобільність тощо) заняття за рахунок самостійної роботи.

Процедура оскарження результатів контрольних заходів оцінювання. Студент може підняти будь-яке питання, яке стосується процедури контрольних заходів та очікувати, що воно буде розглянуто згідно із наперед визначеними процедурами Положенням про апеляцію на результати підсумкового семестрового контролю знань студентів Чернівецького національного університету імені Юрія Федьковича (<https://www.chnu.edu.ua/media/h0fn0fgh/polozhennia-pro-apeliatsiiu.pdf>).

Інклюзивне навчання. Засвоєння знань та умінь в ході вивчення дисципліни «Технології .NET розробки мобільних аплікацій» може бути доступним для більшості осіб з особливими освітніми потребами, окрім здобувачів з серйозними вадами зору, які не дозволяють виконувати завдання за допомогою персональних комп'ютерів, ноутбуків та/або інших технічних засобів.

Навчання іноземною мовою. У ході виконання завдань студентам може бути рекомендовано звернутися до англomовних джерел.

Підготовка до практичних занять та контрольних заходів здійснюється під час самостійної роботи студентів з можливістю консультування з викладачем у визначений час консультацій або за допомогою електронного листування (електронна пошта, месенджери).

Критерії оцінювання результатів навчання з навчальної дисципліни

Підсумковий контроль із дисципліни «Технології .NET розробки мобільних аплікацій» здійснюється у формі заліку, який підсумовує всі результати навчальної діяльності студента протягом семестру. Залік виставляється як загальна підсумкова оцінка з дисципліни, яка складається з суми балів за результатами поточно-модульного контролю (ПМК) та захисту підсумкового проекту.

Якщо студент на заліку отримав незадовільну оцінку, то це вважається як академічна заборгованість і набрані бали не заносяться до відомості. За графіком деканату студент перескладає залік і його результати заносяться до окремої відомості.

Під час заліку перевіряється:

- засвоєння студентом фундаментальних концепцій крос-платформної мобільної розробки, архітектури .NET MAUI та патерну MVVM;
- умінь проєктувати та створювати структуровані, функціональні та масштабовані мобільні аплікації;
- здатність реалізовувати повний цикл роботи з даними через Entity Framework Core, включаючи CRUD-операції;
- навички інтеграції систем аутентифікації, авторизації, роботи з API та платформо-специфічними можливостями;
- вміння оптимізувати продуктивність аплікацій, впроваджувати тестування та налагодження;
- здатність презентувати власний проєкт, демонструвати його роботу на різних платформах і пояснювати архітектурні та технологічні рішення.

Оцінка за підсумковий контроль формується на основі суми балів, набраних за поточну роботу та підсумкову перевірку знань, і переводиться у систему ECTS та національну шкалу оцінювання.

Розподіл балів, які отримують студенти

Поточне оцінювання (аудиторна та самостійна робота)	Сумарна к-ть балів
---	--------------------

Змістовий модуль 1					Змістовий модуль 2					100
T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	
10	10	10	10	10	10	10	10	10	10	

Шкала оцінювання: національна та ЄКТС

Оцінка за національною шкалою	Оцінка за шкалою ЄКТС	
	Оцінка (бали)	Пояснення за розширеною шкалою
Відмінно	A (90-100)	відмінно
Добре	B (80-89)	дуже добре
	C (70-79)	добре
Задовільно	D (60-69)	задовільно
	E (50-59)	достатньо
Незадовільно	FX (35-49)	(незадовільно) з можливістю повторного складання
	F (1-34)	(незадовільно) з обов'язковим самостійним опрацюванням освітнього компоненту до перескладання

Перелік питань для самоконтролю та підсумкового контролю навчальних досягнень студентів

1. Проведіть порівняльний аналіз архітектурних підходів Xamarin.Forms та .NET MAUI. Які ключові переваги забезпечує міграція на .NET MAUI?
2. Охарактеризуйте життєвий цикл мобільного застосунку .NET MAUI та проаналізуйте, як події життєвого циклу впливають на управління ресурсами.
3. Порівняйте процес компіляції застосунків для платформ iOS, Android та Windows у .NET MAUI. Які специфічні вимоги існують для кожної платформи?
4. Проаналізуйте структуру проекту .NET MAUI та обґрунтуйте призначення кожного ключового файлу конфігурації (MauiProgram.cs, App.xaml, MainPage.xaml).
5. Дайте змістовну оцінку переваг та недоліків крос-платформної розробки порівняно з нативною розробкою для iOS та Android.
6. Охарактеризуйте інструменти налагодження в Visual Studio для мобільних застосунків. Як вибір емулятора чи фізичного пристрою впливає на процес розробки?
7. Проаналізуйте еволюцію мобільних технологій Microsoft від Windows Phone до .NET MAUI. Які уроки було винесено з попередніх платформ?
8. Порівняйте різні типи контейнерів компоновання (StackLayout, Grid, FlexLayout, AbsoluteLayout) та обґрунтуйте вибір оптимального контейнера для різних сценаріїв UI.
9. Проведіть аналіз принципів Material Design (Google) та Human Interface Guidelines (Apple). Як забезпечити відповідність обом стандартам у крос-платформному застосунку?
10. Охарактеризуйте механізми створення адаптивного дизайну в .NET MAUI. Як забезпечити коректне відображення UI на пристроях з різною щільністю пікселів?
11. Дайте порівняльний аналіз декларативного (XAML) та імперативного (C#) підходів до створення UI. У яких випадках доцільно використовувати кожен з них?
12. Проаналізуйте систему ресурсів та стилів у .NET MAUI. Як організувати ієрархію стилів для забезпечення консистентності дизайну?

13. Охарактеризуйте принципи доступності (Accessibility) у мобільних застосунках. Які техніки підвищують зручність використання для користувачів з обмеженими можливостями?
14. Порівняйте статичні та динамічні ресурси (StaticResource vs DynamicResource). Проаналізуйте вплив на продуктивність та випадки застосування.
15. Проведіть порівняльний аналіз архітектурних патернів MVC, MVP та MVVM у контексті мобільної розробки. Які критерії визначають вибір патерну?
16. Охарактеризуйте принцип розділення відповідальностей (Separation of Concerns) у патерні MVVM. Як це впливає на тестованість коду?
17. Проаналізуйте різницю між одностороннім (OneWay) та двостороннім (TwoWay) зв'язуванням даних. У яких сценаріях кожен тип є оптимальним?
18. Дайте змістовну оцінку механізму INotifyPropertyChanged. Які альтернативні підходи до реалізації реактивності існують у сучасних фреймворках?
19. Порівняйте реалізацію команд через ICommand з традиційною обробкою подій. Які архітектурні переваги забезпечує використання команд?
20. Охарактеризуйте роль Dependency Injection у патерні MVVM. Як DI контейнер спрощує управління залежностями між компонентами?
21. Проаналізуйте переваги використання CommunityToolkit.Mvvm порівняно з ручною реалізацією MVVM. Які можливості надає цей фреймворк?
22. Дайте порівняльний аналіз стратегій навігації у MVVM: через View vs через ViewModel. Яка стратегія забезпечує кращу тестованість?
23. Охарактеризуйте різницю між Custom Controls, User Controls та Templated Controls. Коли доцільно створювати кожен тип?
24. Проаналізуйте механізм DataTemplate та ControlTemplate. Як вони дозволяють відокремити дані від їх візуального представлення?
25. Порівняйте CollectionView та ListView у .NET MAUI. Які критерії продуктивності визначають вибір компонента для відображення списків?
26. Дайте змістовну оцінку різних типів навігації (Shell, NavigationPage, TabbedPage, FlyoutPage). Які архітектурні наслідки має кожен підхід?
27. Охарактеризуйте Animation API у .NET MAUI. Як анімації впливають на користувацький досвід та продуктивність застосунку?
28. Проаналізуйте систему розпізнавання жестів. Як забезпечити консистентну поведінку жестів на різних платформах?
29. Порівняйте можливості Microsoft.Maui.Graphics з нативними API малювання на iOS та Android. Які обмеження існують у крос-платформному підході?
30. Проведіть порівняльний аналіз SQLite, Realm та LiteDB для мобільних застосунків. Які фактори визначають вибір бази даних?
31. Охарактеризуйте підходи Code First та Database First в Entity Framework Core. Яка стратегія є оптимальною для мобільної розробки?
32. Проаналізуйте механізм міграцій у Entity Framework Core. Як забезпечити безпечне оновлення схеми бази даних на пристроях користувачів?
33. Дайте змістовну оцінку стратегій завантаження зв'язаних даних (Eager Loading, Lazy Loading, Explicit Loading). Як вибір стратегії впливає на продуктивність?
34. Порівняйте паттерни Repository та Unit of Work. Як вони спрощують тестування та підтримку коду доступу до даних?
35. Охарактеризуйте особливості асинхронної роботи з базою даних у мобільних застосунках. Як async/await впливає на відгук інтерфейсу?
36. Проаналізуйте стратегії синхронізації локальних та віддалених даних. Як вирішити конфлікти при одночасному редагуванні на різних пристроях?
37. Порівняйте різні протоколи аутентифікації (OAuth 2.0, OpenID Connect, SAML). Які переваги має OAuth 2.0 для мобільних застосунків?

38. Охарактеризуйте потоки автентифікації (Authentication Flows) у OAuth 2.0. Який потік є найбезпечнішим для мобільних клієнтів?
39. Проаналізуйте механізми безпечного зберігання облікових даних на мобільних пристроях. Порівняйте SecureStorage, Keychain (iOS) та Keystore (Android).
40. Дайте змістовну оцінку структури та захисту JWT токенів. Які вразливості можуть виникнути при неправильній реалізації?
41. Порівняйте підходи Session-based та Token-based аутентифікації. Чому токени є кращим вибором для мобільних застосунків?
42. Охарактеризуйте процес інтеграції з зовнішніми провайдерами ідентифікації (Google, Facebook, Microsoft). Які переваги та ризики несе Social Login?
43. Проаналізуйте стратегії оновлення токенів доступу (Refresh Tokens). Як забезпечити безперервну роботу застосунку без повторного входу?
44. Дайте порівняльний аналіз моделей авторизації: Role-Based Access Control (RBAC) vs Attribute-Based Access Control (ABAC).
45. Охарактеризуйте різницю між виконанням ML моделей на пристрої (on-device) та у хмарі. Які критерії визначають вибір підходу?
46. Проаналізуйте можливості Azure Cognitive Services для мобільних застосунків. Порівняйте Computer Vision, Text Analytics та Speech Services.
47. Дайте змістовну оцінку ML.NET у контексті мобільної розробки. Які обмеження існують при тренуванні моделей на мобільних пристроях?
48. Порівняйте архітектурні підходи до реал-тайм комунікації: WebSockets, Server-Sent Events, SignalR. Які переваги надає SignalR?
49. Охарактеризуйте протоколи транспортування в SignalR (WebSockets, Server-Sent Events, Long Polling). Як SignalR вибирає оптимальний протокол?
50. Проаналізуйте стратегії масштабування SignalR застосунків. Як використання Azure SignalR Service вирішує проблеми горизонтального масштабування?
51. Дайте порівняльний аналіз патернів обробки помилок у реал-тайм з'єднаннях. Як забезпечити стійкість до мережевих збоїв?
52. Охарактеризуйте архітектуру Handlers у .NET MAUI. Як вони замінили Custom Renderers з Xamarin.Forms?
53. Проаналізуйте різницю між Platform Handlers та Effects. У яких випадках доцільно використовувати кожен підхід?
54. Порівняйте можливості .NET MAUI Essentials з прямим викликом нативних API. Які компроміси існують між зручністю та гнучкістю?
55. Дайте змістовну оцінку механізмів умовної компіляції для різних платформ. Як мінімізувати кількість платформи-специфічного коду?
56. Охарактеризуйте процес створення Bindings для нативних бібліотек. Які виклики виникають при інтеграції Objective-C та Java коду?
57. Проаналізуйте систему дозволів (Permissions) у .NET MAUI. Як забезпечити коректну обробку дозволів на різних версіях Android та iOS?
58. Порівняйте підходи до реалізації Push-нотифікацій: Firebase Cloud Messaging, Apple Push Notification Service, Azure Notification Hubs.
59. Дайте порівняльний аналіз можливостей доступу до сенсорів пристрою (акселерометр, гіроскоп, магнітометр). Які сценарії використання для кожного сенсора?
60. Охарактеризуйте інструменти профілювання для .NET MAUI застосунків. Порівняйте можливості Visual Studio Profiler, Xamarin Profiler та Android Studio Profiler.
61. Проаналізуйте типові причини витоків пам'яті у мобільних застосунках. Як паттерни Weak Reference та Event Aggregator допомагають їх уникнути?
62. Дайте змістовну оцінку стратегій віртуалізації у CollectionView. Як recycling patterns впливають на використання пам'яті?
63. Порівняйте підходи до кешування даних: in-memory cache, disk cache, distributed cache. Коли кожен підхід є оптимальним?

64. Охарактеризуйте техніки оптимізації мережевих запитів. Проаналізуйте ефективність compression, batching та request coalescing.
65. Проаналізуйте різницю між паралелізмом (Parallelism) та конкурентністю (Concurrency) у контексті мобільної розробки. Коли використання багатопотоковості обґрунтоване?
66. Дайте порівняльний аналіз стратегій завантаження зображень. Як бібліотеки типу FFImageLoading оптимізують роботу з графікою?
67. Охарактеризуйте метрики продуктивності для мобільних застосунків (CPU usage, memory consumption, battery drain, network traffic). Як Application Insights допомагає моніторити ці метрики?
68. Проведіть порівняльний аналіз пірамід тестування для мобільних застосунків. Яке співвідношення Unit/Integration/UI тестів є оптимальним?
69. Охарактеризуйте відмінності між фреймворками xUnit, NUnit та MSTest. Які можливості кожного роблять його кращим для певних сценаріїв?
70. Проаналізуйте паттерни організації тестів: AAA (Arrange-Act-Assert), Given-When-Then. Як ці паттерни покращують читабельність тестів?
71. Дайте змістовну оцінку техніки Test-Driven Development (TDD) у контексті мобільної розробки. Які переваги та виклики несе цей підхід?
72. Порівняйте підходи до мокування залежностей: ручне створення моків vs використання фреймворків (Moq, FakeItEasy, NSubstitute).
73. Охарактеризуйте виклики UI-тестування мобільних застосунків. Як інструменти типу Appium вирішують проблеми крос-платформного тестування?
74. Проаналізуйте стратегії логування у виробничих застосунках. Порівняйте структуроване логування (Serilog) з традиційними підходами.
75. Дайте порівняльний аналіз CI/CD платформ для мобільної розробки (GitHub Actions, Azure DevOps, App Center). Які фактори визначають вибір платформи?
76. Проведіть комплексний аналіз життєвого циклу розробки мобільного застосунку від ідеї до публікації в магазинах. Які критичні точки прийняття рішень існують на кожному етапі?
77. Охарактеризуйте архітектуру сучасного enterprise мобільного застосунку. Як поєднати MVVM, Clean Architecture, Dependency Injection та Domain-Driven Design?
78. Проаналізуйте еволюцію крос-платформних фреймворків (PhoneGap, Ionic, React Native, Flutter, .NET MAUI). Які тренди визначають майбутнє мобільної розробки?
79. Дайте змістовну оцінку компромісів між нативною та крос-платформною розробкою у 2025 році. Чи існують сценарії, де .NET MAUI є однозначно кращим вибором?
80. Порівняйте підходи до забезпечення безпеки мобільних застосунків на різних рівнях: мережа, зберігання даних, аутентифікація, код. Як створити комплексну стратегію безпеки?

Зарахування результатів неформальної освіти

Зарахування результатів неформальної освіти під час вивчення навчальної дисципліни «**Технології .NET розробки мобільних аплікацій**» здійснюється відповідно до *Положення про взаємодію формальної та неформальної освіти у Чернівецькому національному університеті імені Юрія Федьковича* (<https://www.chnu.edu.ua/media/3aykf41y/polozhennia-pro-vzaємodіiu-formalnoi-ta-neformalnoi-osvity.pdf>).

Визнання результатів навчання, здобутих у межах неформальної освіти, спрямоване на підтримку академічної мобільності студентів, розвиток їхньої самостійності, творчості та професійної компетентності.

До форм неформальної освіти, результати якої можуть бути враховані при вивченні дисципліни, належать:

- проходження професійно орієнтованих онлайн-курсів, вебінарів, тренінгів і майстер-класів;
- участь у навчальних проєктах, воркшопах і практиках;

- виконання індивідуальних навчальних проєктів у межах відкритих освітніх платформ (Prometheus, Coursera, Udemy, Stepik, EdEra, Codecademy, FreeCodeCamp тощо).

При наявності *сертифікатів або офіційних підтверджень* про проходження неформального навчання за тематикою, що відповідає змісту дисципліни, студенту може бути *перезараховано до 35% змістового матеріалу курсу*.

Таке зарахування можливе за умови *підготовки студентом*:

1. Презентації або аналітичної доповіді за матеріалами прослуханого курсу / тренінгу.
2. Коротких письмових нотаток (рефлексії) із викладенням основних результатів та практичних висновків.
3. Публічного захисту (презентації) результатів перед групою під час практичного заняття.

У межах дисципліни «Технології .NET розробки мобільних аплікацій» неформальна освіта також може виступати як *форма індивідуального науково-дослідного завдання (ІНДЗ)*.

Студентам пропонується проходження онлайн-курсів або мікропрограм за одним із

Результати проходження курсів оцінюються у *межах поточного контролю* як виконання індивідуального завдання та можуть становити *до 10 балів за один модуль*, залежно від рівня завершення:

- 9-10 балів – проходження курсу на 80-100% із виконанням фінального проєкту або тесту;
- 8 балів – проходження курсу на 60-79%;
- 7 балів – проходження курсу на 40-59%;
- 6 балів і нижче – часткове опрацювання без підтвердження сертифікатом.

Оцінювання підтверджується поданням:

- електронного сертифіката або іншого офіційного документа про завершення курсу;
- короткого звіту (1–2 стор.) із власним аналізом отриманих знань та їх зв'язку зі змістом дисципліни.

Усі матеріали неформальної освіти проходять *верифікацію викладачем* відповідно до академічної доброчесності та тематики дисципліни.

Рекомендована література:

Основна

1. Del Sole A. .NET MAUI Succinctly : електрон. кн. Morrisville : Syncfusion, 2024. 118 p. URL: <https://www.syncfusion.com/succinctly-free-ebooks/dotnet-maui-succinctly> (дата звернення: 14.02.2026).
2. Eubanks M. .NET MAUI in Action. Shelter Island : Manning Publications, 2023. 400 p. URL: <https://www.manning.com/books/dotnet-maui-in-action> (дата звернення: 14.02.2026).
3. Britch S. .NET MAUI documentation : вебсайт / Microsoft Learn. 2026. URL: <https://learn.microsoft.com/en-us/dotnet/maui/> (дата звернення: 14.02.2026).
4. Olson C. Real-World .NET MAUI: Build cross-platform desktop and mobile apps from scratch. Birmingham : Packt Publishing, 2022. 512 p.
5. Ortinau D., Versluis G. .NET MAUI for Beginners : відеокурс та матеріали / Microsoft Learn. URL: <https://learn.microsoft.com/en-us/shows/dotnet-maui-for-beginners/> (дата звернення: 14.02.2026).
6. Mattone R. Mastering .NET MAUI: Build Multi-platform Native UI with High Performance. Birmingham : Packt Publishing, 2023. 380 p.
7. Microsoft Learn. Build apps with .NET MAUI : навч. шлях. 2026. URL: <https://learn.microsoft.com/en-us/training/paths/build-apps-with-dotnet-maui/> (дата звернення: 14.02.2026).
8. Versluis G. .NET MAUI Samples : репозиторій прикладів коду / GitHub. URL: <https://github.com/dotnet/maui-samples> (дата звернення: 14.02.2026).
9. Davis J. Enterprise Application Patterns Using .NET MAUI : електрон. кн. Microsoft Press, 2022. URL: <https://learn.microsoft.com/en-us/dotnet/architecture/maui/> (дата звернення: 14.02.2026).

10. Lakshen V. .NET MAUI Projects: Build 8 real-world cross-platform apps with .NET MAUI. Birmingham : Packt Publishing, 2023. 450 p.

Додаткова (допоміжна)

1. Lawrence S. Introducing .NET MAUI: Build and Deploy Cross-Platform Applications Using C# and .NET 9.0 Multi-Platform App UI. 2nd ed. Berkley : Apress, 2025. 412 p. *(Найновіше видання з фокусом на .NET 9 та Visual Studio Code)*.
2. Krämer A. Develop cross-platform apps with .NET MAUI. 2nd ed. [S. l.] : Self-published, 2024. 540 p. *(Грунтовне оновлення 2024 року, що охоплює локалізацію, кастомні контроли та розгортання в сторах)*.
3. Versluis G. Mastering .NET MAUI: Build high-performance native apps for iOS, Android, macOS, and Windows. 2nd ed. Birmingham : Packt Publishing, 2025. 450 p. *(Книга від головного інженера команди MAUI, очікується/виходить у 2025 році)*.
4. Del Sole A. .NET MAUI Succinctly. Morrisville : Syncfusion, 2024. 120 p. URL: <https://www.syncfusion.com/succinctly-free-ebooks/dotnet-maui-succinctly> (дата звернення: 14.02.2026).
5. Eubanks M. .NET MAUI in Action. Shelter Island : Manning Publications, 2023. 400 p. *(Практичний посібник із фокусом на архітектурі та реальних кейсах)*.
6. Mattone R. Mastering .NET MAUI: Build Multi-platform Native UI with High Performance. Birmingham : Packt Publishing, 2023. 382 p.
7. Davis J. Enterprise Application Patterns Using .NET MAUI : електрон. кн. Microsoft Learn, 2024. URL: <https://learn.microsoft.com/en-us/dotnet/architecture/maui/> (дата звернення: 14.02.2026). *(Оновлено у 2024 році для роботи з архітектурою eShop)*.
8. Lakshen V. .NET MAUI Projects: Build 8 real-world cross-platform apps with .NET MAUI. Birmingham : Packt Publishing, 2023. 450 p. *(Збірник проєктів: від месенджерів до ігор)*.
9. Olson C. Real-World .NET MAUI: Build cross-platform desktop and mobile apps from scratch. Birmingham : Packt Publishing, 2022/2023. 512 p.
10. Britch S. .NET MAUI Documentation : електрон. посіб. / Microsoft Learn. 2026. URL: <https://learn.microsoft.com/en-us/dotnet/maui/> (дата звернення: 14.02.2026). *(Жива документація, актуалізована під .NET 10)*.

Інформаційні ресурси

1. Документація .NET Multi-platform App UI (.NET MAUI) : вебсайт / Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/maui/> (дата звернення: 14.02.2026).
2. Навчальний шлях «Build mobile and desktop apps with .NET MAUI» : вебсайт / Microsoft Learn. URL: <https://learn.microsoft.com/en-us/training/paths/build-apps-with-dotnet-maui/> (дата звернення: 14.02.2026).
3. Офіційний блог .NET (розділ MAUI) : вебсайт / Microsoft. URL: <https://devblogs.microsoft.com/dotnet/category/maui/> (дата звернення: 14.02.2026).
4. Репозиторій .NET MAUI на GitHub : вебсайт. URL: <https://github.com/dotnet/maui> (дата звернення: 14.02.2026).
5. .NET MAUI Community Toolkit : бібліотека та документація / GitHub. URL: <https://github.com/CommunityToolkit/Maui> (дата звернення: 14.02.2026).
6. Awesome .NET MAUI : кураторський список бібліотек та ресурсів / J. Suarez Ruiz. URL: <https://github.com/jsuarezruiz/awesome-dotnet-maui> (дата звернення: 14.02.2026).
7. Матеріали конференції .NET Conf 2025 : вебсайт / Microsoft. URL: <https://www.dotnetconf.net/> (дата звернення: 14.02.2026).
8. Серія відеоуроків «.NET MAUI for Beginners» : вебсайт / YouTube Channel «dotnet». URL: <https://www.youtube.com/playlist?list=PLdo4fOcmZ0oUBAdL2NwBpJIs3GKEfaoz> (дата звернення: 14.02.2026).

9. Портал для розробників Syncfusion .NET MAUI : навчальні матеріали та контроли. URL: <https://www.syncfusion.com/blogs/category/maui> (дата звернення: 14.02.2026).
10. Спільнота Reddit r/dotnetMAUI : форум розробників. URL: <https://www.reddit.com/r/dotnetMAUI/> (дата звернення: 14.02.2026).

Політика академічної доброчесності

Політика академічної доброчесності у межах навчальної дисципліни «Технології .NET розробки мобільних аплікацій» ґрунтується на засадах, визначених *Етичним кодексом Чернівецького національного університету імені Юрія Федьковича* (<https://www.chnu.edu.ua/media/xe1lulcg/etychnyi-kodeks-chernivetskoho-natsionalnoho-universytetu.pdf>).

Здобувачі вищої освіти зобов'язані дотримуватися принципів чесності, об'єктивності, прозорості, відповідальності та поваги до інтелектуальної власності. Усі види навчальної діяльності – виконання індивідуальних завдань, створення вебпроектів, підготовка презентацій, тестування, модульний і підсумковий контроль – мають здійснюватися самостійно, із дотриманням авторства та належним посиланням на використані джерела.

Будь-які прояви академічної недоброчесності, зокрема плагіат, фабрикація чи фальсифікація даних, списування, необґрунтоване отримання допомоги від інших осіб, подання чужих робіт як власних або повторне використання вже оцінених робіт без погодження з викладачем, розцінюються як порушення академічних норм. Такі випадки підлягають розгляду відповідно до процедур, визначених університетським *Етичним кодексом*, із можливим застосуванням академічної відповідальності – від повторного виконання завдання до анулювання результатів поточного або підсумкового контролю.

Викладачі, зі свого боку, зобов'язані забезпечувати об'єктивність і неупередженість під час оцінювання, створювати умови для добросовісного навчання, а також сприяти формуванню у студентів культури академічної відповідальності та усвідомленого дотримання норм етики.

Таким чином, політика академічної доброчесності у межах дисципліни спрямована на формування високої культури академічного спілкування, професійної відповідальності та поваги до інтелектуальних результатів інших осіб, що є запорукою якісної освіти та академічної репутації університету.