



# Силабус дисципліни «ТЕХНОЛОГІЇ РОЗРОБКИ ANDROID-ЗАСТОСУНКІВ»

**Викладач:** Бурлаченко Іван Сергійович  
старший викладач кафедри комп'ютерної інженерії  
факультету комп'ютерних наук ЧНУ імені Петра Могили.

## Очікувані результати навчання

В результаті вивчення дисципліни студент  
*має знати:*

- принципи побудови архітектури мобільних застосунків, які здійснюються шляхом визначення цілей розроблення ПЗ, вхідних і вихідних даних, декомпозиції програмних систем для Android та iOS, на підсистеми, компоненти або модулі та розроблення структури користувацького інтерфейсу.
- основні патерни архітектури програмних систем на базі Android та iOS, що приймаються групою архітекторів і аналітиків;
- особливості організації розробки мобільних програмних продуктів невеликими групами розробників;
- організаційні етапи передачі мобільних програмних продуктів для публікації маркетингових online-платформах Play Market и App Store.
- як розділяти програмні системи на множини незалежних задач, доступних для розуміння і розв'язання;
- як впорядкувати та організувати складові частин проблеми в ієрархічні структури з додаванням нових деталей на кожному рівні.
- як абстрагувати та відокремлювати істотні аспекти системи й нехтувати несуттєвими абстракціями;
- як формалізувати загальне методологічне вирішення проблеми;

**Обсяг:** буде визначено після здійснення процедури вибору студентами вибіркових дисциплін.

**Мета:** навчитися використовувати системний підхід до проектування архітектури мобільних програмних продуктів, працювати в командах mobile розробників.

## Оригінальність навчальної дисципліни

Матеріали дисципліни представляють сукупний досвід інженерії програмного забезпечення багатьох tech-лідерів проєктів та solution-архітекторів міжнародних ІТ компаній.

Студент *має вміти*

- обґрунтовувати й узгоджувати елементи мобільних програмних систем;
- використовувати засоби тестування і розробки програми для платформи Android та iOS;
- естетично компоувати елементи управління, що призначені для побудови користувацького інтерфейсу, використовувати графічні служби і механізми спливаючих вікон для Android та iOS застосунків;
- працювати зі службами визначення місця розташування Google Maps і OpenStreetMap;
- застосовувати елементи управління для відтворення музичних файлів, відеофайлів і інших засобів мультимедіа;
- працювати з даними акселерометру і іншими датчиками;
- використовувати фреймворки(каркаси) для розробки ігор і анімації;
- знати методи для постійного зберігання даних в файлах і вбудованих базах даних;
- отримувати доступ до веб-служб GraphQL, RESTful та працювати з JSON та іншими форматами даних;
- тестувати в цілому мобільні програмні системи та окремі компоненти.

## Зміст дисципліни

Тема 1. Специфіка операційної системи Android. Архітектура Android API

- 1) SDK/NDK, IDE, емулятори.
- 2) Структура проекту. Gradle. Аспекти роботи Android застосунків з GitLab.
- 3) Клас Application як точка входу в застосунок.
- 4) Життєвий цикл застосунку. Fragment. Task.

Тема 2. Основні компоненти програмної системи на Android. Використання ресурсів та активів у Android застосунках.

- 1) Context. Manifest. Intent.
- 2) Activity. Service. ContentProvider. BroadCast (Receiver)
- 3) Рядки (локалізація). Розміри. Растрові та векторні зображення.
- 4) Стили. View/ViewGroup. Клас View. Клас ViewGroup.
- 5) Короткий розбір найбільш популярних спадкоємців View, ViewGroup.

Тема 3. Особливості процесу створення мобільних Android застосунків.

- 1) XML-верстка. Рендеринг xml, взаємодія з елементами UI.
- 2) Action Bar. Men. Dialog. Навігація і взаємодія екранів.
- 3) Поглиблений розбір можливостей Intent. Взаємодія Activity. Fragment.
- 4) BackStack, навігація, передача даних. Списки. Адаптери.
- 5) ListView, GridView. ViewHolder. RecyclerView.

Тема 8 Гібридні мобільні фреймворки на основі популярних мов програмування

- 1) Flutter. Архітектура фреймворку з урахуванням Dart. Види компонування UI елементів. Панелі дій та вкладок. Діалогові вікна та компоненти.
- 2) Kivy. Архітектура фреймворку з урахуванням Python. Види компонування UI елементів. Панелі дій та вкладок. Діалогові вікна та компоненти.
- 3) Xamarin. Архітектура фреймворку. Види компонування UI елементів.

Тема 4. Асинхронне програмування в Android. Взаємодія з ОС.

- 1) Thread. Runnable. Handler. AsyncTask. Локальні дані. SharedPreferences.

2) Робота з пам'яттю. Робота з базами даних (SQLite). Restful / GraphQL. Нативні інструменти. Сервіси. Intent Service.

- 3) Основні принципи, HTTP / HTTPS-протоколи. JSON. Broadcast Receiver. Content Provider. Планувальник, оповіщення, widgets. Популярні в Android поведінкові/породжуючі патерни проектування SOLID, MVP, MVVM, MVC.
- 4) Огляд популярних бібліотек: Gson, Picasso / Glide. Retrofit2. ORM на прикладі Room (Google Architecture components)

Тема 5. Вивчення мови Swift. Основні відомості про мову. Синтаксичні парадигми.

- 1) Оператори. Рядки і символи. Колекції.
- 2) Керуючі структури ООП в Swift. Властивості і методи. Спадкування та інкапсуляція.
- 3) Життєвий цикл об'єктів. Механізми управління пам'яттю. Протоколи, категорії.

Тема 6. Робота з ресурсами у Swift. Робота з колекціями.

- 1) Рядки, локалізація. Зображення. Робота з колекціями. Набори (Set). Масиви (Array)
- 2) Словники (Dictionary). Потоки (Thread). Черги (Queues).
- 3) Вкладені типи. Розширення. Протоколи. Універсальні шаблони.
- 4) Автоматичний підрахунок посилань (ARC). Безпека зберігання. Контроль доступу. Просунуті оператори.

Тема 7. JavaScript-орієнтовані гібридні мобільні фреймворки

- 1) React Native Компоненти навігації та маршрутизації Види компонування UI елементів. Панелі дій та вкладок. Діалогові вікна та компоненти.
- 2) Ionic Компоненти навігації та маршрутизації Види компонування UI елементів. Панелі дій та вкладок. Діалогові вікна та компоненти.
- 3) Swelte Native. Компоненти навігації та маршрутизації Утиліти шаблонізації компонентів. Види компонування UI елементів. Панелі дій та вкладок. Діалогові вікна та компоненти.

### Пререквізити

Програмування, об'єктно-орієнтоване програмування, алгоритми і структури даних

### Постреквізити

інженерія програмного забезпечення, крос-платформне програмування, web-технології.

**Семестровий контроль:** залік/іспит

**Оцінювання:**

За семестр: 70/60 балів

За залік/іспит: 30/40 балів

**Види робіт:**

Лабораторні роботи – 60/50 балів (12/10 x 5 б)

Індивідуальне проєктне завдання – 10 балів.

**Технічне забезпечення**

Компілятор мови Go

Android Studio, XCode, VS Code IDE

Moodle, Google Meet, Zoom, Webex, профілі соціальних мереж

**Політика щодо дедлайнів**

Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку.

**Політика щодо академічної доброчесності**

Передбачає самостійне виконання лабораторних робіт та індивідуального проєктного завдання. Списування під час заліку (в т. ч. із використанням мобільних пристроїв) заборонено. У разі виявлення плагіату або списування роботи не зараховуються.

**Критерії оцінювання лабораторних робіт**

- вміння самостійно вирішувати поставлені задачі та представляти рішення у вигляді оптимального програмного коду.
- вміння користуватися специфічними інструментальними засобами для створення програмного забезпечення, що задовольняє SOLID принципам.
- вміння лаконічно та логічно формувати відповіді на запитання, пов'язані з виконаними лабораторними роботами.

**Критерії оцінювання індивідуального проєктного завдання**

- формулювання актуальності, проблеми, мети і завдань, практичного значення;
- вміння самостійно створювати оптимальний програмний код для вирішення поставленої задачі з використанням патернів програмування.;
- вміння користуватися специфічними інструментальними засобами для розробки ПЗ з ефективною архітектурою;
- якість доповіді студента (форма доповіді, зміст, доказова база, висновки) ; повнота та логічність відповідей на поставлені питання.

