

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

ЗАТВЕРДЖЕНО  
на засіданні кафедри  
інформаційних систем  
Протокол № 1 від 22.08.2023 р.

ПОГОДЖЕНО  
Проректор з навчально-методичної роботи



Каріна НЕМАШКАЛО

**МОБІЛЬНІ ТЕХНОЛОГІЇ**  
робоча програма навчальної дисципліни (РПНД)

Галузь знань 12 "Інформаційні технології"  
Спеціальність 121 "Інженерія програмного забезпечення"  
Освітній рівень перший (бакалаврський)  
Освітня програма "Інженерія програмного забезпечення"

Статус дисципліни  
Мова викладання, навчання та оцінювання

вибіркова  
українська

Розробник:  
к.т.н., доцент

підписано КЕП

Андрій ПОЛЯКОВ

Завідувач кафедри  
інформаційних систем

Дмитро БОНДАРЕНКО

Гарант програми

Олег ФРОЛОВ

Харків  
2024

## ВСТУП

Навчальна дисципліна "Мобільні технології" є ключовою дисципліною, що охоплює концепції та практики, які стосуються розробки мобільних додатків. З урахуванням високого рівня проникнення мобільних пристроїв у суспільство та їх впливу на наш повсякденний життя, знання та навички, пов'язані з розробкою мобільних додатків, стали критично важливими для інформаційних технологій. Дисципліна зорієнтована на вивчення мови програмування Kotlin та Android SDK як основних інструментів для створення додатків на платформі Android. Здобувачі отримують розуміння процесу розробки від початкового проектування до тестування та випуску додатку. Також докладно вивчається інтерфейс користувача, взаємодія з сенсорами пристрою, робота з мережею, зберігання даних, використання сервісів та багато іншого. Окрім того, ця дисципліна закладає основи архітектури додатків, щоб забезпечити надійне та продуктивне програмне забезпечення. Здобувачі отримують практичний досвід роботи з проектами, що сприятиме кращому засвоєнню теоретичного матеріалу.

Вивчення дисципліни "Мобільні технології" передбачає освоєння навичок розробки мобільних додатків, зокрема: вміння працювати з Android Studio та SDK Android, розуміння та використання мови програмування Kotlin, побудова інтерфейсу додатку з використанням XML, використання SQLite для збереження даних в додатку, інтеграція з веб-службами через HTTP і REST, робота з мультимедійними додатками в Android, використання Google Maps та використання системи керування версіями Git. Окрім того, важливо набути вміння розробляти додатки, що відповідають сучасним стандартам безпеки, продуктивності та кращим практикам дизайну.

Метою викладання навчальної дисципліни "Мобільні технології" є надання здобувачам вищої освіти системи спеціальних теоретичних знань та практичних навичок у сфері розробки мобільних додатків на платформі Android. Розвиток аналітичних вмінь, навичок роботи в команді, використання сучасних інструментів, середовищ розробки та методологій для розробки бізнес-орієнтованих мобільних додатків. Вивчення навчального матеріалу дозволяє розвивати у здобувачів навички роботи в сфері мобільних технологій, адаптуватися до умов, що швидко змінюються на ринку мобільних додатків і технологій.

Завданнями навчальної дисципліни є:

- надання здобувачам необхідних теоретичних знань про архітектуру платформи Android, мову програмування Kotlin і різні інструменти та бібліотеки для розробки Android додатків;
- розвиток практичних навичок розробки мобільних додатків для ОС Android, включаючи проектування UI/UX, кодування на Kotlin, використання баз даних та веб-служб, роботу з мультимедіа та геолокаційними сервісами;
- навчання здобувачів використанню SDK Android, середовища розробки Android Studio і системи контролю версій Git для ефективної командної роботи;

- обізнаності здобувачів із найновішими трендами та найкращими практиками у сфері розробки мобільних додатків;
- розвиток навичок аналізу та вирішення проблем, що виникають під час розробки додатків;
- підготовка здобувачів для ефективної роботи в команді та самостійного виконання проектів;
- ознайомлення здобувачів із процесом розгортання та тестування мобільних додатків на різних пристроях та платформах;
- виховання в здобувачів відповідального ставлення до якості коду і продуктивності додатків.

Предметом навчальної дисципліни є набір знань і вмінь, потрібних для ефективного проектування, кодування, тестування та оптимізації мобільних додатків на платформі Android. Це включає вивчення мови програмування Kotlin, розуміння архітектури платформи Android, використання інструментів та бібліотек для розробки Android-додатків, а також працюючи з базами даних, сервісами, мережею і камерою смартфона.

Об'єктом навчальної дисципліни є процеси розробки, тестування і внесення змін до мобільних додатків, створених для функціонування на платформі Android.

Результати навчання та компетентності, які формує навчальна дисципліна визначено в табл. 1.

Таблиця 1

### **Результати навчання та компетентності, які формує навчальна дисципліна**

Результати навчання	Компетентності, якими повинен оволодіти здобувач вищої освіти
PH07	ЗК05, СК13.
PH12	СК03, СК14.
PH17	ЗК02, СК12.
PH18	ЗК06, СК12.
PH21	СК08.

де, PH07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

PH12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.

PH17. Вміти застосовувати методи компонентної розробки програмного забезпечення.

PH18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

PH21. Знати, аналізувати, вибирати, кваліфіковано застосовувати засоби забезпечення інформаційної безпеки (в тому числі кібербезпеки) і цілісності даних відповідно до розв'язуваних прикладних завдань та створюваних програмних систем.

ЗК02. Здатність застосовувати знання у практичних ситуаціях.

ЗК05. Здатність вчитися і оволодівати сучасними знаннями.

ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

СК03. Здатність розробляти архітектури, модулі та компоненти програмних систем.

СК08. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення

СК12. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.

СК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

СК14. Здатність до алгоритмічного та логічного мислення.

## **ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

### **Зміст навчальної дисципліни**

**Змістовий модуль 1. Середовище розробника для мобільних пристроїв. Мова Kotlin.**

**Тема 1. Введення. Архітектура та компоненти мобільної платформи Android. Середовище розробника. GIT.**

**1.1.** Вступ до дисципліни. Контрольні заходи та результати навчання.

**1.2.** Огляд Android та його архітектури: вступ до Android: історія, версії та розробники Android; архітектура Android (ядро Linux, бібліотеки, Android Runtime, Application Framework); основні компоненти Android (Activity, Services, Broadcast Receivers, Content Providers); огляд Android Studio та його основних функцій; створення простого додатка за допомогою Android Studio.

**1.3.** Вступ до Git та його основні команди: що таке Git та чому його використовують? встановлення Git та настройка середовища; створення нового репозиторію Git та клонування існуючого; створення комітів, гілок та перегляд історії комітів; робота з віддаленими репозиторіями (push, pull, fetch);

**1.4.** Поняття про Android App Bundle та APK; що таке APK та як його створити; що таке Android App Bundle та його відмінність від APK; створення та перевірка Android App Bundle; використання Split APK для розповсюдження додатків; оптимізація розміру APK та Android App Bundle.

**1.5.** Дебагінг Android додатку: вступ до дебагінгу в Android Studio; налаштування точок переривання (breakpoints); використання окна Debug для відстеження виконання коду; дебагінг при використанні підключених пристроїв; відлагодження проблем з пам'яттю та витоку пам'яті за допомогою Android Studio.

**1.6.** Використання Logcat для моніторингу поведінки програми: вступ до Logcat і його завдання; використання різних рівнів викликів Log (Log.d, Log.i, Log.w, Log.e, Log.v, Log.wtf); фільтрація виходу Logcat за допомогою тегів і рівнів важливості; запис та експорт виводу Logcat; використання Logcat для розбору виключень та помилок.

**1.7.** Android Gradle та його використання: вступ до системи збірки Gradle; основні конфігураційні файли Gradle: build.gradle (project) та build.gradle (module); додавання залежностей до проекту Android; розуміння сценаріїв

Gradle та циклу життя збірки Gradle; використання Gradle для створення APK та Android App Bundles.

## **Тема 2. Основи мови Kotlin. Базові конструкції. Лямбда. Класи. Ідіоми.**

**2.1.** Вступ до Kotlin та його базові конструкції: вступ до Kotlin: історія, особливості та переваги; перша програма на Kotlin; робота зі змінними: var і val; базові типи даних та їх оператори; контроль потоку: if-else, switch, цикли.

**2.2.** Функції та лямбда-вирази в Kotlin: створення та виклик функцій; використання параметрів функції та визначення типів повернення; використання простих та блочних лямбда-виразів; лямбда-вирази з приймачами та функціональні типи; колекції та високорівневі функції: map, filter, reduce.

**2.3.** Класи та об'єкти в Kotlin: створення та використання класу та його об'єктів; створення конструкторів та використання властивостей; наслідування в Kotlin: super-класи та перевизначення; абстрактні класи, інтерфейси та їх реалізація, делегування; обмеження видимості (private, protected, internal, public); внутрішні (nested) та внутрішні (inner) класи.

**2.4.** Використання null-безпеки в Kotlin: основи null-безпеки в Kotlin; робота зі змінними, які можуть бути null; використання оператора безпеки "??", оператора "Elvis" "?:", оператора "!!" та його наслідки.

**2.5.** Ідіоми Kotlin та хороші практики: вступ до Kotlin ідіом; робота з нульовими значеннями (null safety); використання ключових слів "apply", "let", "it", "also" та "run"; використання операторів "with" та "lazy"; деструктивні оголошення та копіювання.

## **Тема 3. Мова Kotlin. Колекції, фільтрація, перетворення. Виключення.**

**3.1.** Робота з колекціями в Kotlin: створення і використання списків (List), множин (Set) і словників (Map); операції над колекціями: додавання, видалення, пошук елементів; незмінні та мутовані колекції; створення і використання mutable колекцій; проходження по колекції за допомогою циклу for; використання колекцій в якості параметрів функцій.

**3.2.** Робота зі списками в Kotlin: створення списків, додавання та вилучення елементів; опрацювання списків: цикл for, методи map, forEach; операції зі списками (сортування, видалення дублікатів); перетворення списків (toSet(), toMap()); конвертація списків в рядки і навпаки.

**3.3.** Фільтрація та перетворення колекцій в Kotlin: фільтрація колекцій за допомогою функції filter; перетворення елементів колекції використовуючи функцію map; вбудовані методи для пошуку в колекціях (find, findLast, firstOrNull, lastOrNull); використання послідовності декількох операцій через chaining; використання flatMap і flatten для роботи з вкладеними колекціями; використання агрегуючих функцій (reduce, fold, max, min, sum, average тощо).

**3.4.** Робота з виключеннями в Kotlin: що таке виключення, і навіщо їх обробляти; генерування виключень за допомогою throw; обробка виключень за допомогою блоків try/catch; блок finally і його використання; створення власних виключень.

**3.5.** Створення та використання розширень (extensions) в Kotlin: розширення в Kotlin: що це та як їх використовувати; створення розширювальних функцій; створення розширювальних властивостей; використання розширень в реальному кодї; робота з розширеннями сторонніх класів.

## **Змістовий модуль 2. Архітектура та засоби розробки для ОС Android**

### **Тема 4. Розмітка екрана: основи XML. Елементи екрану, контейнери елементів.**

**4.1.** Основи XML для розмітки екрана Android: вступ до мови XML; використання XML в Android для розмітки екрана; синтаксис XML для опису елементів екрана; базові атрибути XML для елементів у Android; розмітка екрана у Android Studio з використанням XML.

**4.2.** Типи елементів на екрані в Android: використання текстових полів (TextView); робота з кнопками (Button); Створення списків за допомогою RecyclerView; введення тексту через EditText; відображення зображень через ImageView.

**4.3.** Використання контейнерів для групування елементів на екрані: використання LinearLayout для розмітки екрана; робота з RelativeLayout для створення більш складних інтерфейсів; застосування FrameLayout для позиціювання елементів на екрані; GridLayout та його особливості; ConstraintLayout для створення гнучких інтерфейсів.

**4.4.** Адаптація розмітки екрана для різних розмірів пристроїв та орієнтацій: реагування на зміну орієнтації екрана; створення різних розміток для портретної та ландшафтної орієнтацій; використання ConstraintLayout для адаптивної верстки; використання Android Studio Preview для перегляду адаптивної верстки; робота з різними розмірними ресурсами.

**4.5.** Кастомізація елементів екрана та стилізація інтерфейсу: зміна властивостей елементів: кольорів, шрифтів, та інші атрибути; створення та використання стилів та тем; візуальна стилізація кнопок, списків та інших елементів; використання векторних та растрових зображень; анімування елементів до інтерфейсу.

### **Тема 5. Доступ та керування елементами екрана з коду. Опрацювання подій.**

**5.1.** Доступ до елементів екрана з коду: використання методу findViewById для отримання доступу до елементів; використання Android Kotlin Extensions для автоматичної генерації об'єктів відображень; робота з properties елементів екрана; обробка винятків при роботі з елементами розмітки; використання LiveData для спостереження за станом UI компонентів.

**5.2.** Інтерактивні елементи управління – Button, EditText, Checkbox та інші: впровадження кнопки і обробка натискань на кнопку; використання текстового поля вводу (EditText); використання прапорців та перемикачів (CheckBox, Switch); робота з випадаючими списками (Spinner); працюючи з Radio Button та Radio Group.

**5.3.** Події взаємодії користувача: обробка подій клацання (click events); обробка подій торкання (touch events); обробка подій long click; обробка подій клавіатури; обробка подій перемикання фокусу.

**5.4.** Робота з перелічувачами (RecyclerView): основи роботи із RecyclerView; створення Adapter для управління елементами RecyclerView; обробка подій натискання на елементи списку RecyclerView; оптимізація продуктивності RecyclerView; реалізація складних списків з допомогою RecyclerView.

**5.5.** Використання анімацій для покращення UX: основи системи анімації Android; анімація зміни властивостей елементів (ObjectAnimator); просторові анімації переходів (Transition); анімація переходів між Activity (Activity Transition); використання Animated Vector Drawables для створення складних анімацій.

**Тема 6. Application, Activity та її життєвий цикл. Manifest. Система дозволів на Android.**

**6.1.** Загальні аспекти роботи з Application та Activity в Android: загальна структура Android-застосунку; роль та стратегії використання класу Application; концепція та роль Activity в Android-застосунку; створення нових Activity та інтеграція їх до застосунку; методи для запуску та завершення Activity.

**6.2.** Життєвий цикл Activity: розуміння життєвого циклу Activity; обробка початкових стадій життєвого циклу (onCreate, onStart); реагування на паузу та відновлення Activity (onPause, onResume); обробка завершення життєвого циклу Activity (onStop, onDestroy); використання savedInstanceState для збереження стану Activity.

**6.3.** Компонент Android Manifest в застосунках: розуміння ролі та використання Android Manifest файлу; декларування Activity в манифесті; використання манифесту для контролю орієнтації екрана та інших налаштувань Activity; декларування опційного доступу до апаратного забезпечення та служб; використання манифесту для декларування дозволів (permissions).

**6.4.** Робота з дозволами в Android: розуміння концепції дозволів в Android; вимоги до дозволів, окреслені в манифесті Android; запит дозволів під час виконання; обробка сценаріїв відмови або прийняття дозволів користувачем; рекомендації щодо найкращих практик управління дозволами.

**6.5.** Intent та робота з ними в рамках Activity: розуміння ролі інтентів в Android; створення та використання інтентів для запуску Activity; використання Intent'ів для передачі даних між Activity; реагування на Intent'и в Activity; використання Intent-фільтрів для реагування на системні події.

**Тема 7. Fragment та його життєвий цикл. Android Jetpack Components: navigation.**

**7.1.** Загальні аспекти роботи з Fragment в Android: призначення і основні властивості Fragment; створення нового Fragment; інтеграція Fragment в Activity; взаємодія між Fragment і Activity; динамічне додавання, видалення або заміна Fragment в Activity.

**7.2.** Життєвий цикл Fragment: загальний огляд життєвого циклу Fragment; обробка створення Fragment: методи onAttach, onCreate, onCreateView; обробка

переходу Fragment в стан активності: методи `onActivityCreated`, `onStart`, `onResume`; обробка призупинення та зупинки Fragment: методи `onPause`, `onStop`; обробка знищення View і самого Fragment: методи `onDestroyView`, `onDestroy`, `onDetach`.

**7.3.** Робота з бібліотекою Android Jetpack та компонентами навігації: загальне знайомство з бібліотекою Android Jetpack та архітектурними компонентами; вступ до Navigation Component; організація навігації між Fragment з допомогою Navigation Component; обробка та передача даних між Fragment з допомогою Navigation Component; реагування на системні події, такі як кнопка "Back", в рамках навігації.

**7.4.** Розробка гнучких UI за допомогою Fragment: розробка адаптивного UI з використанням Fragment; компонування декількох Fragment в одному Activity; створення вкладок або посувних екранів за допомогою Fragment; створення діалогових вікон чи вікон повноекранних рекламних повідомлень за допомогою DialogFragment; використання Fragment для побудови навігаційних елементів меню.

**7.5.** Оптимізація роботи з даними в Fragment: використання ViewModel для управління даними в Fragment; зберігання і відновлення стану Fragment; використання архітектурних компонентів для роботи з базами даних в Fragment; повторне використання об'єктів View та логіки відображення в різних Fragment; використання LiveData для спостереження за даними в Fragment.

**Тема 8. Зберігання даних. Робота з файлами. Preference. Основи баз даних, Room.**

**8.1.** Зберігання даних у файлі: введення в механізм зберігання даних в файл; зберігання інформації у внутрішній пам'яті пристрою; зберігання інформації на зовнішній картоці пам'яті; читання і запис даних в файл; безпека при зберіганні даних в файл.

**8.2.** Робота з Shared Preferences: визначення та робота з Shared Preferences; запис даних в SharedPreferences; читання даних із SharedPreferences; редагування та видалення даних з SharedPreferences; збереження складних структур даних з SharedPreferences.

**8.3.** Введення в роботу з базами даних SQLite: використання SQLite в Android; створення бази даних та створення таблиць; вставка, оновлення і видалення рядків; запити до бази даних: вибірка й сортування даних; закриття бази даних і обробка виключень.

**8.4.** Використання ORM-бібліотеки Room: введення та налаштування бібліотеки Room; створення бази даних і сутностей за допомогою Room; створення DAO для доступу до бази даних; вставка, оновлення та видалення записів за допомогою Room; створення запитів та обробка результату.

**8.5.** Синхронізація з віддаленою базою даних: робота з віддаленими API за допомогою Retrofit; застосування Gson для обробки JSON відповідей; робота з допомогою OkHttp; використання паттерну Repository для подальшої асинхронної роботи з базою даних; використання LiveData для спостереження за змінами в базі даних.



### **Змістовий модуль 3. Бібліотеки та розширенні засоби розробки для ОС Android**

#### **Тема 9. Вихід до Інтернету. Retrofit, JSON.**

**9.1.** Вступ у доступ до Інтернету та HTTP запитів: перегляд сценаріїв використання Інтернету у Android; вибір мережевих правил доступу; використання Handler та Looper для вирішення задач мережі; використання HttpURLConnection для запитів до Інтернету; застосування AsyncTask для мережних запитів.

**9.2.** Загальний огляд і використання Retrofit: вступ до Retrofit; налаштування Retrofit для виконання HTTP запитів; конфігурація API End-Point'ів та будівничих запитів з Retrofit; використання Retrofit для синхронних та асинхронних викликів; обробка відповідей та помилок з Retrofit.

**9.3.** Робота з API та JSON: ознайомлення з форматом JSON; subscribe End-Point'и API, що повертають JSON; десеріалізація JSON конвертерами Retrofit; серіалізація даних у JSON для надсилання на сервер; обробка складних JSON відповідей.

**9.4.** Інтеграція OkHttp з Retrofit: використання OkHttp як HTTP-клієнта для Retrofit; журналювання HTTP запитів та відповідей за допомогою OkHttp; налаштування виконання запитів з додатковими заголовками або параметрами; управління непередбаченістю мережі та повторенням запиту; обробка відповідей з кодами помилки HTTP за допомогою OkHttp.

**9.5.** Бібліотека Gson для обробки JSON: вступ до бібліотеки Gson; переведення JSON до об'єктів Java за допомогою Gson; перетворення об'єктів Java на JSON за допомогою Gson; використання анотацій Gson для точного відображення JSON; обробка вкладених й масивних об'єктів у JSON через Gson.

#### **Тема 10. Основи архітектури додатку. MVVM, ViewModel, LiveData, coroutine.**

**10.1.** Введення в архітектуру додатку: важливість добре структурованої архітектури додатку; загальний огляд принципів SOLID; загальний огляд патерну MVC; загальний огляд патерну MVP; загальний огляд патерну MVVM.

**10.2.** Застосовуємо патерн MVVM: приклад реалізації патерну MVVM на Kotlin; визначення ViewModel та її ролі в архітектурі MVVM; визначення View та її ролі в архітектурі MVVM; визначення Model та її ролі в архітектурі MVVM; комунікація між компонентами в MVVM.

**10.3.** Використання ViewModel для збереження та управління даними UI: вступ до класу ViewModel і його використання; створення ViewModel для активності або фрагмента; збереження та відновлення даних через ViewModel; визначення життя ViewModel в контексті циклу життя власника; використання ViewModel з LiveData.

**10.4.** Використання LiveData для реактивного програмування: вступ до LiveData і його використання у Android; створення і використання LiveData об'єктів; інтеграція LiveData з ViewModel; оновлення значень LiveData і спостереження за змінами; використання Transformations з LiveData.

**10.5.** Застосування coroutines для асинхронних завдань: вступ до coroutines у Kotlin; порівняння coroutines та callbacks/AsyncTask; створення та

запуск простих coroutines; управління багатопоточністю з coroutines: Dispatchers, withContext; обробка помилок і виключні ситуації в контексті coroutines.

## **Тема 11. Services, WorkManager. Notifications.**

**11.1.** Розуміння та використання служб (services): вступ до служб Android; створення простої служби (Service); створення зв'язаної служби (Bound Service); запуск і зупинка служби; синхронізація даних у фоновому режимі за допомогою служб.

**11.2.** Робота з WorkManager: вступ до WorkManager; створення простого завдання (WorkRequest); запуск завдання (WorkRequest); збереження стану та обробка результатів завдань; управління нестабільним з'єднанням і використання періодичних запитів.

**11.3.** Введення в сповіщення: розуміння та переваги сповіщень; створення і правильне налаштування каналу сповіщень; створення та трансляція сповіщення; відкриття активності через сповіщення; налаштування дій та кнопок в сповіщенні.

**11.4.** Створення постійних сповіщень: загальні ідеї та використання постійних повідомлень; створення повідомлення, яке не можна видалити або відкласти; налаштування аудіо та інтенсивних параметрів повідомлень; додавання великого зображення або тексту до сповіщення; посилання повідомлення, яке вимагає миттєвого реагування користувача.

**11.5.** Робота з Foreground services: введення в foreground services; створення foreground service; переведення service у foreground режим; оновлення сповіщення foreground service; зупинка або закінчення foreground service.

## **Тема 12. Робота з GPS, робота з картами, робота з камерою.**

**12.1.** Використання GPS та отримання геолокації: вступ до GPS та служб геолокації в Android; запит дозволів на геолокацію; отримання поточного місцезнаходження користувача; робота з об'єктом Location; лістенири зміни місцезнаходження.

**12.2.** Робота з картами за допомогою Google Maps API: вступ у Google Maps API; додавання Google Map до вашої програми; налаштування та використання об'єкта GoogleMap; додавання маркерів та креслень на карту; рух камери та контроль перегляду.

**12.3.** Робота з камерою: вступ до використання камери в Android; запит дозволів на доступ до камери; запуск Intetn'a камери для фотографування; отримання та збереження фотографії; використання Android Camera API для більш детального контролю.

**12.4.** Підключення до і використання Google Play Services: загальні знання про Google Play Services; встановлення та підключення Google Play Services до проекту; визначення та вирішення проблем зі зв'язком; використання геолокаційних служб Google Play Services; використання інших API Google Play Services, таких як Google Drive API, Google Fit API, тощо.

**12.5.** Розширений контроль камери за допомогою Camera2 API: вступ в API Camera2; робота з об'єктом CameraDevice; створення запиту на попередній

перегляд; налаштування об'єкта CaptureRequest для фотографування; працюючи з вмиканням/вимиканням спалаху, фокусуванням та іншими налаштуваннями камери.

Перелік лабораторних занять за навчальною дисципліною наведено в табл. 2.

Таблиця 2

### Перелік лабораторних занять

Назва теми	Зміст
Тема 1 – 2	Дослідження інструментальних засобів розроблення мовою Kotlin. Базові конструкції мови Kotlin. Контроль версій коду GIT
Тема 3	Дослідження ООП та лямбда в мові Kotlin
Тема 4 – 6	Розгортання середовища розробника Android Studio та дослідження базових елементів Activity, життєвого циклу, та між-компонентної комунікації
Тема 5 – 7	Дослідження використання RecyclerView списків у мобільному додатку та використанням фрагментів та навігації між ними
Тема 8	Дослідження роботи баз даних на Android за допомогою Room Persistence Library
Тема 9	Дослідження роботи веб-службами за допомогою Retrofit і обробка JSON-відповідей
Тема 10 – 11	Створення асинхронних сервісів з використанням Kotlin Coroutines та LiveData з ViewModel. Зберігання та передача стану Activity при повороті пристрою.
Тема 12	Дослідження роботи Goole Map, GPS та робота з камерою

Перелік самостійної роботи за навчальною дисципліною наведено в табл. 3.

Таблиця 3

### Перелік самостійної роботи

Назва теми	Зміст
Тема 1 – 12	Вивчення лекційного матеріалу
Тема 1 – 12	Підготовка до лабораторних занять
Тема 1 – 12	Підготовка до поточних контрольних робіт
Тема 1 – 12	Підготовка до екзамену

Кількість годин лекційних та лабораторних занять, а також годин самостійної роботи наведено в робочому плані (технологічній карті) з навчальної дисципліни.

## МЕТОДИ НАВЧАННЯ

У процесі викладання навчальної дисципліни для набуття визначених результатів навчання, активізації освітнього процесу передбачено застосування таких методів навчання, як:

Словесні (лекція (Тема 1 – 3, 4, 7, 9, 12), проблемна лекція (Тема 1, 4, 6 – 8), лекція-візуалізація (Тема 1 – 12)).

Наочні (демонстрація (Тема 1 – 12)).

Лабораторна робота (Тема 1 – 12), кейс-метод (Тема 4, 6 – 8, 9 – 12).

## ФОРМИ ТА МЕТОДИ ОЦІНЮВАННЯ

Університет використовує 100 бальну накопичувальну систему оцінювання результатів навчання здобувачів вищої освіти.

**Поточний контроль** здійснюється під час проведення лекційних, лабораторних занять і має на меті перевірку рівня підготовленості здобувача вищої освіти до виконання конкретної роботи і оцінюється сумою набраних балів:

– для дисциплін з формою семестрового контролю **екзамен (іспит):** максимальна сума – 60 балів; мінімальна сума, що дозволяє здобувачу вищої освіти скласти **екзамен (іспит) – 35 балів.**

**Підсумковий контроль** включає семестровий контроль та атестацію здобувача вищої освіти.

**Семестровий контроль** проводиться у формі семестрового екзамену (іспиту). Складання семестрового екзамену (іспиту) здійснюється під час екзаменаційної сесії.

Максимальна сума балів, яку може отримати здобувач вищої освіти під час екзамену (іспиту) – 40 балів. Мінімальна сума, за якою екзамен (іспит) вважається складеним – 25 балів.

**Підсумкова оцінка за навчальною дисципліною** визначається сумуванням балів за поточний та підсумковий контроль.

Під час викладання навчальної дисципліни використовуються наступні контрольні заходи:

Поточний контроль: захист лабораторних робіт (40 балів), поточні контрольні роботи (10 балів), презентацій (10).

Семестровий контроль: Екзамен (40 балів)

Більш детальну інформацію щодо системи оцінювання наведено в робочому плані (технологічній карті) з навчальної дисципліни.

Приклад екзаменаційного білета та критерії оцінювання для навчальної дисципліни.

## Приклад екзаменаційного білета

Харківський національний економічний університет імені Семена Кузнеця

Перший (бакалаврський) рівень вищої освіти

Спеціальність «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні системи та технології»

Семестр I

Навчальна дисципліна «Мобільні технології»

### ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 1

#### Завдання 1. Тест.

##### Питання № 1

**Яке твердження, у файлі build.gradle, вірно позначає, що відповідний модуль - це модуль бібліотеки Android?**

Виберіть один з 4 варіантів відповіді:

- |    |                                      |
|----|--------------------------------------|
| 1) | apply plugin: 'com.module.library'   |
| 2) | apply plugin: 'com.android.library'  |
| 3) | apply plugin: 'com.module.library'   |
| 4) | include plugin: 'com.module.library' |

##### Питання № 2

**Чого немає у життєвому циклі активності?**

Виберіть один з 4 варіантів відповіді:

- |    |            |
|----|------------|
| 1) | onPause()  |
| 2) | onResume() |
| 3) | onOpen()   |
| 4) | onStart()  |

##### Завдання №3

**True чи false? Коли ваш додаток переходить у фоновий режим, не гарантується його знищення. Він може лише зупинитися і чекати на повернення користувача.**

Виберіть один з 2 варіантів відповіді:

- |    |       |
|----|-------|
| 1) | True  |
| 2) | False |

##### Питання № 4

**True чи false? Коли використовується атрибут enabled, він визначає, чи є view видимим..**

Виберіть один з 2 варіантів відповіді:

- |    |       |
|----|-------|
| 1) | False |
| 2) | True  |

### Питання № 5

**Який правильний синтаксис для виведення "Hello World" на мові Kotlin?**

Виберіть один з 4 варіантів відповіді:

- |    |   |
|----|---|
| 1) | <code>println("Hello World")</code>               |
| 2) | <code>Console.WriteLine("Hello World")</code>     |
| 3) | <code>cout &lt;&lt; "Hello World";</code>         |
| 4) | <code>System.out printline("Hello World");</code> |

### Питання № 6

**Як почати писати цикл **\*\*while\*\*** у Kotlin?**

Виберіть один з 4 варіантів відповіді:

- |    |                                  |
|----|----------------------------------|
| 1) | <code>while (x &lt; y)</code>    |
| 2) | <code>if x &gt; y while</code>   |
| 3) | <code>while x &lt; y</code>      |
| 4) | <code>while x &lt; y then</code> |

### Питання № 7

**Що є `to` у прикладі нижче:**

`kotlin`

`val test = 33 to 42`

Виберіть один з 4 варіантів відповіді:

- |    |   |
|----|---|
| 1) | Ключовим словом Kotlin для створення <code>Pair(33, 42)</code>            |
| 2) | Помилкою синтаксису   |
| 3) | Ключовим словом Kotlin для створення <code>Range</code> від 33 до 42      |
| 4) | Інфіксьним розширювальним методом, який створює <code>Pair(33, 42)</code> |

### Питання № 8

**Яке оголошення функції є правильним у Kotlinі?**

Виберіть один з 4 варіантів відповіді:

- |    |  |
|----|--|
| 1) | <code>int sum(int a, int b)</code>             |
| 2) | <code>function sum(a: Int, b: Int): Int</code> |
| 3) | <code>int sum(a: Int, b: Int)</code>           |
| 4) | <code>fun sum(a: Int, b: Int): Int</code>      |

### Питання № 9

**Вкажіть варіант, який використовується для обробки нульових винятків у Kotlin?**

Виберіть один з 4 варіантів відповіді:

- |    |                              |
|----|------------------------------|
| 1) | <code>Range</code>           |
| 2) | <code>Sealed Class</code>    |
| 3) | <code>Elvis Operator</code>  |
| 4) | <code>Lambda function</code> |

## Питання № 10

### Яка поведінка за замовчуванням у класів Kotlin?

Виберіть один з 4 варіантів відповіді:

1)	Усі класи public
2)	Усі класи sealed
3)	Усі класи final
4)	Усі класи abstract

### Завдання 2. (евристичне)

Розробити мобільний додаток який складається з однієї екранної форми і відображає інформацію про один об'єкт відповідної предметної області. Додаток повинен завантажувати та зберігати дані у файл.

Додаток повинен забезпечувати (використовувати):

макет з обмеженнями який дає змогу створювати складні користувацькі інтерфейси за допомогою обмежень (constraints) між різними елементами;

обмежувати віртуальну клавіатуру у відповідності до даних що вводяться;

зберігання та відновлення даних при переході у фоновий режим та використовуючи збереження даних у внутрішнє файлове сховище для постійного зберігання у форматі серіалізації JSON;

викликати коротке Toast-повідомлення щодо збереження даних;

за допомогою кнопки з звичайним утриманням (Click) зробити відкриття веб-сторінки в браузері;

інтернаціоналізацію: англійська та українська мови;

використовувати систему версіонування коду GIT. Усі реалізації вимог оформити у вигляді фіксацій (commit). Завантажити (push) на server gitlab.hneu.net у проект **examen**, що створено в акаунті.

Затверджено на засіданні кафедри

інформаційних систем протокол № \_\_\_ від « \_\_\_ » \_\_\_\_\_ 20\_\_ р.

Екзаменатор

\_\_\_\_\_

к.т.н., доц. Поляков А. О.

Зав. кафедрою

\_\_\_\_\_

к.т.н., доц. Бондаренко Д. О.

### Критерії оцінювання

Підсумкові бали за екзамен складаються із суми балів за виконання всіх завдань, що округлені до цілого числа за правилами математики.

Алгоритм вирішення кожного завдання включає окремі етапи, які відрізняються за складністю, трудомісткістю та значенням для розв'язання завдання. Тому окремі завдання та етапи їхнього розв'язання оцінюються окремо одне від одного таким чином:

#### Завдання 1 (тест).

Перше питання присвячене перевірці теоретичних знань з дисципліни. На тест виноситься 10 питань по 1.5 балу за кожне. Загалом 15 балів. Тривалість тесту 20 хвилин.

#### Завдання 2 (евристичне).

Друге питання присвячене розробці мобільного додатку відносно поставленої задачі та навести програмний проект з програмним кодом та звіт зі скріншотами, що демонструють роботу додатку. Проект розробляється на мові програмування Kotlin та платформі Android. Здобувач повинен створити проект в середовище Android Studio. Основна мета вирішення цієї задачі – перевірка практичних вмінь здобувача проектувати архітектуру додатку, проектувати інтерфейс користувача та вміти реалізувати бізнес-логіку у застосунку. При цьому здобувачу дозволяється користування існуючою довідковою літературою. Після перевірки програми здобувач отримує  $K_1$  балів за наступними вимогами (табл. 4). Друге завдання оцінюється від 0 до 25 балів.

Таблиця 4

#### Критерії оцінювання за діагностичним завданням

Бали	Критерій
25	Завдання виконане в повному обсязі.
24	Завдання виконане в повному обсязі. Є невеликі зауваження до організації структури коду та інтерфейсу.
22-23	Завдання в основному виконано. Є зауваження до організації структури коду та інтерфейсу.
20-21	Завдання виконане, але не в повному обсязі. Програма працює, але не реалізовані деякі вимоги або можливості, зазначені в завданні.
18-19	Завдання виконане, але не в повному об'ємі. Програма працює, але не реалізовані дві або три вимоги або можливості, зазначені в завданні.
16-17	Завдання виконане, але не в повному об'ємі. Програма працює, але не реалізовані чотири або п'ять вимог або можливостей, зазначені в завданні.
11-15	Завдання виконане, але не в повному об'ємі. Програма працює, але не реалізовані більш ніж п'ять вимог або можливостей, зазначені в завданні.
7-10	Програма як мінімум дозволяє правильно виконати одну дію (можливість).
3-6	Програма запускається, але містить грубі помилки, повністю не виконано жодного завдання, вимоги.
2	Програма не відповідає постановці завдання.
1	Програма не містить програмного коду, розробленого здобувачем. Програма має явні ознаки несамостійності її розробки.
0	Програма відсутня.

## РЕКОМЕНДОВАНА ЛІТЕРАТУРА

### Основна

1. Griffiths D. Head first Kotlin / D. Griffiths, D. Griffiths. — Beijing Boston Farnham Sebastopol Tokyo : O'Reilly, 2019. — 448 p.
2. Griffiths D. ead First Android Development: A Learner's Guide to Building Android Apps with Kotlin / D. Griffiths, D. Griffiths. — Beijing Boston Farnham Sebastopol Tokyo : O'Reilly, 2021. — 890 p.



3. Sills B. Android programming: The Big Nerd Ranch guide / B. Sills, B. Gardner, K. Marsicano, C. Stewart. — Atlanta, GA : Big Nerd Ranch, 2022. — 667 p.

4. Поляков А. О Аналіз методів і технологій розроблення мобільних додатків для платформи Android : навчальний посібник [Електронний ресурс] / Поляков А. О, Федорченко В. М, Шматко О. В. — Харків : ХНЕУ ім. С. Кузнеця, 2017. — 286 p. — URL: <http://repository.hneu.edu.ua/handle/123456789/20105>

#### **Додаткова**

5. Jemerov D. Kotlin in Action / D. Jemerov, S. Isakova. — Shelter Island, NY : Manning Publications Co, 2017. — 334 p.

6. Sedunov A. Kotlin In-Depth. [Vol-I]. A Comprehensive Guide to Modern Multi-Paradigm Language. — New Delhi, India : BPB Publications, 2020. — 340 p.

7. Laurence P.-O. Programming Android with Kotlin: Achieving Structured Concurrency with Coroutines / P.-O. Laurence, A. Hinchman-Dominguez, G. V. Meike, M. Dunn. — Beijing Sebastopol, CA : O'Reilly, 2022. — 336 p.

8. Tigcal J. Simplifying Android Development with Coroutines and Flows / J. Tigcal. — Birmingham, UK : Packt Publishing, Limited, 2022. — 159 p.

9. Muschko B. Gradle in action / B. Muschko. — Shelter Island, NY : Manning, 2014. — 456 p.

10. Tuominen T. RxJava for Android Developers / T. Tuominen. — Shelter Island, NY : Manning Publications Co., 2019. — 514 p.

11. Федорченко В. Аналіз ефективності технологій розроблення мобільних застосунків для ОС Android / В. Федорченко, А. Поляков, О. Северінов // Вісник ХНАДУ — 2022. — Вип. 96. — Р. 81–90.

#### **Інформаційні ресурси**

12. Kotlin Docs | Kotlin [Електронний ресурс] // Kotlin Help. — Електрон. дані. — Режим доступу: <https://kotlinlang.org/docs/home.html>.

13. Download Android Studio & App Tools - Android Developers [Електронний ресурс] // Електрон. дані. — Режим доступу: <https://developer.android.com/studio>.

14. Android Compose Tutorial | Jetpack Compose | Android Developers [Електронний ресурс] // Android Developers. — Електрон. дані. — Режим доступу: <https://developer.android.com/jetpack/compose/tutorial>.

15. Design for Safety | App quality | Android Developers [Електронний ресурс] // Android Developers. — Електрон. дані. — Режим доступу: <https://developer.android.com/quality/privacy-and-security>.

16. Fragments | Android Developers [Електронний ресурс] // Android Developers. — Електрон. дані. — Режим доступу: <https://developer.android.com/guide/fragments>.

17. Fundamentals of testing Android apps | Android Developers [Електронний ресурс] // Android Developers. — Електрон. дані. — Режим доступу: <https://developer.android.com/training/testing/fundamentals>.

18. Services overview | Background work [Електронний ресурс] // Android Developers. — Електрон. дані. — Режим доступу: <https://developer.android.com/develop/background-work/services>.

19. Мобільні технології (6.04.121) [Електронний ресурс] / Розробники Андрій Поляков, Володимир Федорченко // Персональні навчальні системи ХНЕУ ім. С. Кузнеця — Електрон. дані. — Режим доступу: <https://pns.hneu.edu.ua/course/view.php?id=1366>.