

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

ЗАТВЕРДЖЕНО
на засіданні кафедри
інформаційних систем
Протокол № 1 від 27.08.2024 р.

ПОГОДЖЕНО
Проректор з навчально-методичної роботи
Каріна НЕМАШКАЛО



**ПРОГРАМУВАННЯ ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ
робоча програма навчальної дисципліни (РПНД)**

Галузь знань
Спеціальність
Освітній рівень
Освітня програма

**12 “Інформаційні технології”
122 “Комп’ютерні науки”
перший (бакалаврський)
“Комп’ютерні науки”**

Статус дисципліни

Мова викладання, навчання та оцінювання

вибіркова

українська

Розробник:
к.т.н., доцент

Андрій ПОЛЯКОВ

Завідувач кафедри
інформаційних систем

Дмитро БОНДАРЕНКО

Гарант програми

Дмитро ГОЛУБНИЧИЙ

Харків
2024

ВСТУП

Навчальна дисципліна "Програмування для мобільних пристроїв" є важливою складовою навчального процесу, яка охоплює основи розробки додатків для мобільних платформ. Враховуючи популярність мобільних пристроїв та їхню роль у сучасному житті, знання в галузі мобільного програмування стають критичними для ІТ-фахівців. Дисципліна фокусується на використанні мови програмування Kotlin та Android SDK, що дозволяє створювати ефективні додатки для Android-платформи. Здобувачі вивчають процес розробки мобільних додатків, починаючи з концептуального проектування, через програмування та тестування, до випуску готового продукту. Особливу увагу приділяють розробці інтерфейсів користувача, інтеграції з сенсорами пристрою, роботі з мережею та зберіганню даних. Також дисципліна надає знання архітектурних підходів до створення мобільних додатків для забезпечення їх надійності та продуктивності. Здобувачі отримують не лише теоретичні знання, але й практичні навички, що допомагають втілювати на практиці концепції, розглянуті на лекціях.

Вивчення дисципліни "Програмування для мобільних пристроїв" передбачає формування практичних навичок розробки мобільних застосунків для платформи Android. У процесі навчання здобувачі опановують роботу з Android Studio та інструментами SDK Android, засвоюють основи мови програмування Kotlin, включно з її базовими конструкціями, класами, лямбда-виразами, ідіомами та колекціями. Значну увагу приділено розробці інтерфейсів користувача з використанням XML, керуванню елементами інтерфейсу через код, а також взаємодії між компонентами програми (Activity, Fragment, Intent). Окремим напрямом є робота з локальним зберіганням даних за допомогою SQLite, SharedPreferences та Room, а також інтеграція з веб-службами через REST API. Здобувачі знайомляться з життєвим циклом додатка, системою дозволів Android, застосуванням анімацій, роботою з Jetpack-компонентами та навігацією. Також розглядаються основи системи контролю версій Git і використання Gradle для збірки проєктів. Особлива увага приділяється питанням відлагодження, забезпечення продуктивності, безпеки та дотриманню кращих практик розробки мобільних застосунків.

Метою викладання навчальної дисципліни «Програмування для мобільних пристроїв» є формування у здобувачів системи спеціалізованих теоретичних знань і практичних навичок у галузі створення мобільних додатків для платформи Android. Особливу увагу приділено розвитку аналітичного мислення, вмінь командної роботи, а також засвоєнню сучасних інструментів, середовищ розробки та методологій, що застосовуються під час реалізації бізнес-орієнтованих мобільних рішень. Засвоєння навчального матеріалу сприяє підготовці здобувачів до професійної діяльності у сфері мобільних технологій і забезпечує їхню здатність швидко адаптуватися до динамічних змін ринку мобільних додатків і технологій.

Завданнями навчальної дисципліни є:

- формування у здобувачів ґрунтовних теоретичних знань щодо архітектури платформи Android, мови програмування Kotlin, а також інструментів і бібліотек, що застосовуються в розробці мобільних додатків;
- розвиток практичних умінь у створенні Android-додатків, зокрема у проєктуванні інтерфейсів користувача (UI/UX), написанні коду на Kotlin, роботі з базами даних, веб-сервісами, мультимедійними можливостями пристроїв і геолокаційними функціями;
- набуття навичок роботи з Android SDK, середовищем розробки Android Studio та системами контролю версій (зокрема Git) для забезпечення ефективної командної взаємодії;
- ознайомлення з актуальними тенденціями та сучасними практиками у сфері мобільної розробки;
- розвиток аналітичного мислення та вміння знаходити рішення технічних і логічних задач, що виникають під час створення додатків;
- підготовка до ефективної співпраці в команді, а також до самостійного виконання програмних проєктів;
- оволодіння методами тестування, налагодження та розгортання мобільних застосунків на різноманітних пристроях і платформах;
- формування відповідального ставлення до якості програмного коду, продуктивності та зручності використання мобільних додатків.

Предметом навчальної дисципліни є сукупність знань і практичних навичок, необхідних для ефективної розробки мобільних додатків на платформі Android. До основних складових входять вивчення мови програмування Kotlin, ознайомлення з архітектурою Android, застосування спеціалізованих інструментів і бібліотек для створення додатків, а також робота з базами даних, сервісами, мережею, камерою смартфона та навігацією.

Об'єктом навчальної дисципліни є процеси проєктування, реалізації, тестування та модифікації мобільних застосунків, що працюють в середовищі 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
ПР05	ЗК12, СК03
ПР09	ЗК06, ЗК12, СК08
ПР10	ЗК11, ЗК12, СК08

ПР15	СК08, СК12, СК14, СК10
------	------------------------

де, 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 застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них із застосуванням мов веб-програмування;

ПР15. Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об’єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

ЗК06. Здатність вчитися й оволодівати сучасними знаннями.

ЗК11. Здатність приймати обґрунтовані рішення.

ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

СК03. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв’язності та нерозв’язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

СК08. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об’єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.

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.

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.Дебаггінг Android додатку: вступ до дебаггінгу в Android Studio; налаштування точок переривання (breakpoints); використання окна Debug для відстеження виконання коду; дебаггінг 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 мutowані 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 коді; робота з розширеннями сторонніх класів.

Змістовий модуль 2. Архітектура та засоби розробки для ОС Android

Тема 1. Розмітка екрана: основи XML. Елементи екрану, контейнери елементів.

1.1.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.

1.2.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.

1.3.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.

1.4.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.

1.5.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 2. Доступ та керування елементами екрана з коду. Опрацювання подій.

2.1.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.

2.2.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 випадючими списками (Spinner); працюючи з Radio Button та Radio Group.

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

2.4.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.

2.5. 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 3. Application, Activity та її життєвий цикл. Manifest. Система дозволів на Android.

3.1. 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.

3.2. 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.

3.3. 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).

3.4. 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.

3.5. 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 4. Fragment та його життєвий цикл. Android Jetpack Components: navigation.

4.1. 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.

4.2. 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.

4.3. 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.

4.4.Розробка гнучких UI за допомогою Fragment: розробка адаптивного UI з використанням Fragment; компонування нескількох 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.

4.5.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 5. Зберігання даних. Робота з файлами. Preference. Основи баз даних, Room.

5.1.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.

5.2.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.

5.3.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.

5.4.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.

5.5.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 1. Вихід до Інтернету. Retrofit, JSON.

1.1.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

URLConnection для запитів до Інтернету; застосування AsyncTask для мережних запитів.

1.2. 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.

1.3. 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.

1.4. 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.

1.5. 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 2. Основи архітектури додатку. MVVM, ViewModel, LiveData, coroutine.

2.1. 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.

2.2. 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.

2.3. 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.

2.4. 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.

2.5. 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 3. Android розробка: Робота з камерою, збереження медіафайлів та дозволи.

3.1. Основи взаємодії з камерою: запит, попередній перегляд та конфігурація: способи доступу до камери; налаштування попереднього

перегляду камери; вибір камери (передня, задня) та обробка випадків відсутності потрібної камери.

3.2.Захоплення фото та відео: процеси та обробка даних; захоплення фотографій з CameraX; запис відео з CameraX; робота з форматами даних (JPEG, MP4, RAW); обробка помилок під час захоплення.

3.3.Управління дозволами: запит та обробка для камери та сховища; декларування необхідних дозволів в AndroidManifest.xml; динамічний запит дозволів під час виконання; найкращі практики роботи з дозволами.

3.4.Збереження медіафайлів: робота зі сховищем та MediaStore; типи сховищ в Android; особливості Scoped Storage (Android 10+); збереження фото та відео за допомогою MediaStore API; збереження у приватне сховище додатка; рекомендації щодо найменування файлів та їх організації.

3.5.Доступ до медіафайлів, обмін та найкращі практики: доступ до існуючих медіафайлів через MediaStore; використання Storage Access Framework; обмін медіафайлами з іншими додатками; оптимізація та життєвий цикл; базові поради щодо тестування функціоналу.

Тема 4. Робота з GPS та картами

4.1.Основи геолокації в Android та робота з GPS: концепція геолокації та її значення в мобільних застосунках; типи провайдерів місцезнаходження; порівняння провайдерів за точністю, швидкістю отримання даних та енергоспоживанням; огляд основних класів та сервісів Android для роботи з геолокацією.

4.2.Отримання даних про поточне місцезнаходження та підписка на оновлення: використання FusedLocationProviderClient для ефективного отримання останнього відомого місцезнаходження; налаштування об'єкта LocationRequest для визначення бажаної точності, інтервалу та частоти оновлень; реалізація підписки на періодичні оновлення місцезнаходження за допомогою requestLocationUpdates() та їх коректне скасування через removeLocationUpdates(); обробка об'єкта Location для отримання координат (широта, довгота), точності, швидкості та часу.

4.3.Управління дозволами для доступу до геолокації та найкращі практики: декларування необхідних дозволів в AndroidManifest.xml; реалізація динамічного запиту дозволів під час виконання (runtime permissions) для Android 6.0 (API 23) та вище; обробка відповідей користувача на запит дозволів; надання чітких пояснень користувачеві про необхідність доступу до геолокації; особливості запиту дозволів для фонового доступу до місцезнаходження на Android 10 (API 29) та вище.

4.4.Інтеграція з картами: Основи роботи з Google Maps SDK: налаштування Google Maps Platform у Google Cloud Console та отримання API ключа; інтеграція Google Maps SDK в Android проект; використання MapView або SupportMapFragment для вбудовування карти в Activity або Fragment; асинхронне завантаження карти та отримання об'єкта GoogleMap через колбек OnMapReadyCallback; базова конфігурація карти: встановлення типу карти.

4.5.Відображення даних на карті, взаємодія та розширені можливості: додавання маркерів (Marker) для позначення конкретних точок: налаштування

позиції, заголовка, сніпета, іконки; кастомізація маркерів та інформаційних вікон; малювання геометричних фігур на карті: ліній (Polyline), полігонів (Polygon), кіл (Circle); керування камерою карти: анімоване переміщення до координат, встановлення рівня зуму, обмеження області перегляду; обробка подій користувача на карті.

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

Таблиця 2

Перелік лабораторних занять

Назва теми	Зміст
Тема 1 -2 Лабораторна робота 1.	Дослідження інструментальних засобів розроблення мовою Kotlin. Базові конструкції мови Kotlin. Контроль версій коду GIT
Тема 3 Лабораторна робота 2.	Дослідження ООП та лямбда в мові Kotlin. Стандартні бібліотеки Kotlin.
Тема 4 – 6 Лабораторна робота 3.	Розгортання середовища розробника Android Studio та дослідження базових елементів Activity, життєвого циклу, та між-компонентної комунікації
Тема 5 – 7 Лабораторна робота 4.	Дослідження механізму навігації та використання списків (RecyclerView) у Android застосунку
Тема 8 Лабораторна робота 5.	Дослідження доступу до даних у Android застосунках що виористовують Room Persistence Library
Тема 9 Лабораторна робота 6.	Створення асинхронних сервісів з використанням Kotlin Coroutines та LiveData з ViewModel.
Тема 10 – 12 Лабораторна робота 7.	Дослідження роботи Goole Map, GPS, та 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 «Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні науки»

Семестр I

Навчальна дисципліна «Програмування для програмних пристроїв»

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 << "Hello World";</code>
4)	<code>System.out.println("Hello World");</code>

Питання № 6	
Як почати писати цикл **while** у Kotlin?	
Виберіть один з 4 варіантів відповіді:	
1)	<code>while (x < y)</code>
2)	<code>if x > y while</code>
3)	<code>while x < y</code>
4)	<code>while x < 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 Pair(33, 42)
2)	Помилкою синтаксису
3)	Ключовим словом Kotlin для створення Range від 33 до 42
4)	Інфіксним розширювальним методом, який створює Pair(33, 42)

Питання № 8	
Яке оголошення функції є правильним у Котліні?	
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)	Range
2)	Sealed Class
3)	Elvis Operator
4)	Lambda function

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.

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 проєкт з програмним кодом та звіт зі скріншотами, що демонструють роботу додатку. Проєкт розробляється на мові програмування Kotlin та платформі Android. Здобувач повинен створити проєкт в середовищі Android Studio. Основна мета вирішення цієї задачі – перевірка практичних вмінь здобувача проєктувати архітектуру додатку, проєктувати 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).

Таблиця 4

Критерії оцінювання за діагностичним завданням

Друге завдання оцінюється від 0 до 25 балів. Кількість балів за друге завдання визначається у відповідності з наступною шкалою:

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 несамотійності 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. Head 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 імені С. Кузнеця, 2017. — 286 p. URL: <http://repository.hneu.edu.ua/handle/123456789/20105>

Додаткова

5. Jemerov D. and others Kotlin in Action / D. Jemerov, S. Isakova. — Shelter Island, NY : Manning Publications Co, 2017. — 334 p.
6. Laurence P.-O. Programming Android with Kotlin: Achieving Structured Concurrency with Coroutines / P.-O. Laurence, A. Hinchman-Dominguez, G. B. Meike, M. Dunn. — Beijing Sebastopol, CA : O'Reilly, 2022. — 336 p.
7. Tigcal J. Simplifying Android Development with Coroutines and Flows / J. Tigcal. — Birmingham, UK : Packt Publishing, Limited, 2022. — 159 p.
8. Muschko B. Gradle in action / B. Muschko. — Shelter Island, NY : Manning, 2014. — 456 p.
9. Tuominen T. RxJava for Android Developers / T. Tuominen. — Shelter Island, NY : Manning Publications Co., 2019. — 514 p.
10. Федорченко В. Аналіз ефективності технологій розроблення мобільних застосунків для ОС Android / В. Федорченко, А. Поляков, О. Сєверінов // Вісник ХНАДУ — 2022. — Вип. 96. — Р. 81–90.

Інформаційні ресурси

11. Kotlin Docs | Kotlin [Електроний ресурс] // Kotlin Help. — Електрон. дані. — Режим доступу: <https://kotlinlang.org/docs/home.html>.
12. Download Android Studio & App Tools - Android Developers [Електроний ресурс] // Електрон. дані. — Режим доступу: <https://developer.android.com/studio>.
13. Android Compose Tutorial | Jetpack Compose | Android Developers [Електроний ресурс] // Android Developers. — Електрон. дані. — Режим доступу: <https://developer.android.com/jetpack/compose/tutorial>.
14. Design for Safety | App quality | Android Developers [Електроний ресурс] // Android Developers. — Електрон. дані. — Режим доступу: <https://developer.android.com/quality/privacy-and-security>.
15. Fragments | Android Developers [Електроний ресурс] // Android Developers. — Електрон. дані. — Режим доступу: <https://developer.android.com/guide/fragments>.
16. Fundamentals of testing Android apps | Android Developers [Електроний ресурс] // Android Developers. — Електрон. дані. — Режим доступу: <https://developer.android.com/training/testing/fundamentals>.
17. Services overview | Background work [Електроний ресурс] // Android Developers. — Електрон. дані. — Режим доступу: <https://developer.android.com/develop/background-work/services>.
18. Програмування для мобільних пристроїв (6.04.122) [Електроний ресурс] / Розробник Андрій Поляков // Персональні навчальні системи ХНЕУ ім. С. Кузнеця — Електрон. дані. — Режим доступу: <https://pns.hneu.edu.ua/course/view.php?id=2170>.