**ЛАБОРАТОРНА РОБОТА №1**

***Тема .*** *Встановлення Android Studio, створення першого проєкту та налаштування GitHub-репозиторію****Мета роботи:*** *Ознайомитися з процесом встановлення Android Studio, створення першого проекту для розробки мобільних додатків на Android та завантаження проєкту на GitHub*

***Теоретичні відомості***

## 1. Android Studio

***Android Studio*** *- це офіційне середовище розробки (IDE) від Google для створення мобільних додатків під Android.  
Основні можливості:*

*🔹* ***Редактор коду на Kotlin/Java*** *з підказками та автодоповненням;  
🔹* ***Візуальний редактор інтерфейсу (Layout Editor)*** *для створення екранів;  
🔹* ***Система збірки Gradle*** *- автоматично компілює код, підключає бібліотеки, створює APK;  
🔹* ***Емулятор Android*** *- дозволяє запускати додаток на віртуальному телефоні;  
🔹* ***Інтеграція з Git*** *- підтримка системи контролю версій прямо з IDE.*

***Структура Android-проєкту:***

* *app/java/... — вихідний код (файли Kotlin/Java);*
* *app/res/layout/... — XML-файли інтерфейсу;*
* *AndroidManifest.xml — опис додатку (назва, права доступу);*
* *build.gradle — налаштування збірки проєкту.*

*Таким чином, Android Studio об’єднує все потрібне для роботи: код, інтерфейс, бібліотеки й тестування.*

## 2. Git

***Git*** *- це система контролю версій, яка дозволяє:*

* *зберігати історію змін у коді;*
* *працювати в команді без «перезаписування» файлів одне одного;*
* *створювати* ***гілки*** *(branches) для окремих завдань;*
* *повертатися до попередніх версій проєкту, якщо сталася помилка.*

*Основні команди Git:*

* *git init - ініціалізувати репозиторій;*
* *git add . - додати файли у staging (перед комітом);*
* *git commit -m "опис" - зафіксувати зміни у локальному репозиторії;*
* *git push - відправити зміни на віддалений сервер (GitHub);*
* *git pull — отримати оновлення з сервера;*
* *git checkout -b feature/lab1 — створити нову гілку.*

## 3. GitHub

***GitHub*** *— це онлайн-платформа, яка зберігає проєкти Git у «хмарі» і дозволяє працювати над ними командно.  
Можливості GitHub:*

* *зберігання коду в публічних або приватних репозиторіях;*
* *створення* ***Pull Request (PR)*** *— коли студент відправляє зміни для перевірки викладачу чи команді;*
* *можливість* ***код-рев’ю****: коментарі до змін, зауваження та виправлення;*
* *відстеження помилок і задач через* ***Issues****;*
* *автоматична перевірка коду через* ***GitHub Actions*** *(CI/CD).*

***Робочий процес у GitHub:***

1. *Студент створює гілку feature/lab1-Прізвище.*
2. *Додає свій проєкт у власну папку (students/Прізвище\_Ім’я/lab1).*
3. *Завантажує зміни на GitHub (git push).*
4. *Створює Pull Request у спільний репозиторій.*
5. *Викладач робить рев’ю і приймає або просить виправлення.*

***Теоретичні запитання***

1. *Що таке Android Studio і для чого вона використовується?*
2. *Які основні компоненти Android-проєкту (поясніть призначення Java/Kotlin, res, AndroidManifest.xml, Gradle)?*
3. *Що таке Activity в Android і яку роль вона відіграє?*
4. *Для чого використовується файл layout (XML) у проєкті Android?*
5. *Яке призначення файлу build.gradle у проєкті Android Studio?*
6. *Що таке SDK і чому воно потрібне для Android-розробки?*
7. *Що таке емулятор Android і для чого він використовується?*
8. *Що таке система контролю версій Git і які її переваги?*
9. *Чим відрізняється локальний репозиторій від віддаленого?*
10. *Які основні команди Git ви знаєте і що вони виконують (init, add, commit, push, pull, checkout)?*
11. *Що таке гілка (branch) у Git і для чого її використовують?*
12. *Що таке GitHub і чим він відрізняється від Git?*
13. *Що таке Pull Request (PR) і яка його роль у командній роботі?*
14. *Що таке .gitignore і для чого він потрібен?*
15. *Як правильно організувати роботу в команді, якщо кілька людей одночасно працюють над одним проєктом у GitHub?*

***Завдання***

*Передумови та позначення*

*ОС: Windows 10/11, macOS, або Linux (Ubuntu/Debian).*

*Потрібні акаунти:*

*GitHub (особистий): https://github.com*

*— увійти/зареєструватись.*

*Позначення, які ви замінюєте на свої:*

***<PRIZVYSCHE>*** *— ваше прізвище латиницею (наприклад, Ferens).*

***<IMYA>*** *— ім’я латиницею (Roman).*

***<ORG>*** *— організація викладача на GitHub (наприклад, saceit-samtech).*

***<COURSE\_REPO>*** *— назва спільного курс-репозиторію (наприклад, mobile-android-2025).*

*Гілка для ЛР1: feature/lab1****-<PRIZVYSCHE>*** *(наприклад, feature/lab1-Ferens).*

*Якщо викладач надасть точні* ***<ORG> і <COURSE\_REPO>,*** *використовуйте їх. Далі в інструкції вказано як діяти у спільному репозиторії (вимога: «на гітхабі створити свою папочку і туди закинути проєкт»).*

***1) Встановлення Android Studio***

*Перейдіть на офіційний сайт: https://developer.android.com/studio*

*Завантажте інсталятор Android Studio (остання стабільна версія).*

*Запустіть інсталятор і залишайте налаштування «за замовчуванням»:*

*Android SDK, Android SDK Platform, Android Virtual Device (AVD/Emulator) — обрати для інсталяції.*

*Після першого запуску Android Studio майстер налаштування докачає компоненти (SDK, Tools).*

*(Windows) Порада щодо емулятора*

*Увімкніть апаратну віртуалізацію у BIOS/UEFI (Intel VT-x/AMD-V).*

*Якщо емультор не стартує: у Windows Features увімкніть Virtual Machine Platform і Windows Hypervisor Platform.*

***2) Встановлення Git та первинні налаштування***

*Windows: завантажити Git для Windows: https://git-scm.com/download/win*

*і встановити.*

*macOS: xcode-select --install або встановити Git з сайту.*

*Linux: sudo apt install git (Debian/Ubuntu).*

*Відкрийте Terminal (або Git Bash у Windows) і налаштуйте ім’я та email (вони відображаються у комітах):*

*git config --global user.name "<IMYA> <PRIZVYSCHE>"*

*git config --global user.email "your\_email@example.com"*

*(Windows) Дозвольте довгі шляхи:*

*git config --global core.longpaths true*

***3) Створення першого Android-проєкту***

*Відкрийте Android Studio → New Project.*

*Шаблон: Empty Activity.*

*Поля заповнити так:*

*Name: CollegeHub-<PRIZVYSCHE> (наприклад, CollegeHub-Ferens).*

*Package name: залишити автоматично згенерований або змінити на ua.saceit.collegehub.<prizvyshche>.*

*Save location: виберіть папку без кирилиці/пробілів у шляху.*

*Language: Kotlin.*

*Minimum SDK: API 24 (Android 7.0) або вище.*

*Натисніть Finish і дочекайтеся індексації Gradle.*

***4) Мінімальна правка інтерфейсу (Hello-екран)***

*В Android Studio відкрийте файл розмітки: app/src/main/res/layout/activity\_main.xml.*

*Замініть вміст на простий «Hello»-екран (можна так):*

*<?xml version="1.0" encoding="utf-8"?>*

*<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"*

*android:orientation="vertical"*

*android:gravity="center"*

*android:background="#FAFAFA"*

*android:layout\_width="match\_parent"*

*android:layout\_height="match\_parent">*

*<TextView*

*android:id="@+id/helloText"*

*android:layout\_width="wrap\_content"*

*android:layout\_height="wrap\_content"*

*android:text="Hello, <PRIZVYSCHE>!"*

*android:textSize="24sp"*

*android:textStyle="bold"*

*android:padding="16dp"/>*

*</LinearLayout>*

*Запустіть застосунок:*

*Створіть AVD: Tools → Device Manager → Create device (наприклад, Pixel 5, API 30+).*

*Натисніть кнопку Run ▶.*

*Переконайтесь, що на екрані емулятора видно текст “Hello, <PRIZVYSCHE>!”.*

*Зробіть скріншот екрану емулятора — він знадобиться у PR. Назвіть файл: lab1-hello-<PRIZVYSCHE>.png.*

***5) Підготовка до роботи зі спільним курс-репозиторієм***

***5.1. Клонування спільного репозиторію курсу***

*На GitHub перейдіть у <ORG>/<COURSE\_REPO> (напр., https://github.com/saceit-samtech/mobile-android-2025).*

*Натисніть Fork (якщо викладач вимагає роботу через форк) або скопіюйте URL для клонування без форку.*

*У терміналі перейдіть у папку, де зберігатимете курс-репозиторії, і виконайте:*

*git clone https://github.com/<ORG>/<COURSE\_REPO>.git*

*cd <COURSE\_REPO>*

*Переконайтеся, що є гілка develop (якщо ні — працюйте з main за вказівкою викладача):*

*git fetch --all*

*git checkout develop || git checkout -b develop*

***5.2. Створення власної гілки для ЛР1***

*git checkout -b feature/lab1-<PRIZVYSCHE>*

***5.3. Структура папок у спільному репозиторії***

*У корені спільного репозиторію створіть свою персональну папку зі стандартизованим шляхом:*

*<PRIZVYSCHE>\_<IMYA>/lab1/*

*Напр.: Ferens\_Roman/lab1/*

***6) Копіювання вашого Android-проєкту у вашу папку курсу***

*Знайдіть локальну папку вашого проєкту CollegeHub-<PRIZVYSCHE> (створеного у розд. 3).*

*Скопіюйте всю папку проєкту (вміст із app, gradle, settings.gradle, build.gradle.kts, gradlew тощо) у:*

*<COURSE\_REPO>/students/<PRIZVYSCHE>\_<IMYA>/lab1/CollegeHub-<PRIZVYSCHE>/*

*Додайте також скріншот емулятора у підпапку:*

*<COURSE\_REPO>/students/<PRIZVYSCHE>\_<IMYA>/lab1/docs/lab1-hello-<PRIZVYSCHE>.png*

*Перевірте, що не завантажуєте сміття: у корені Android-проєкту має бути .gitignore (Android Studio його генерує).*

*Якщо .gitignore відсутній — створіть файл .gitignore і додайте базові правила Android (можна взяти з https://github.com/github/gitignore/blob/main/Android.gitignore*

*).*

***7) Коміт і пуш у віддалений репозиторій курсу***

*Додайте файли до індексу:*

*git add students/<PRIZVYSCHE>\_<IMYA>/lab1/*

*Зробіть коміт із зрозумілим повідомленням (рекомендовано Conventional Commits):*

*git commit -m "feat(lab1): add CollegeHub-<PRIZVYSCHE> project and hello screen"*

*Відправте гілку на GitHub:*

*git push -u origin feature/lab1-<PRIZVYSCHE>*

*Якщо Git просить авторизацію — легше за все увійти в GitHub через Android Studio:*

*File → Settings → Version Control → GitHub → Log In via GitHub.*

***8) Створення Pull Request (PR)***

*Перейдіть у ваш форк або у <ORG>/<COURSE\_REPO> → вкладка Pull requests → New pull request.*

*Base branch: develop (або main, якщо вказав викладач).*

*Compare branch: feature/lab1-<PRIZVYSCHE>.*

*Заголовок PR:*

*Lab1: CollegeHub-<PRIZVYSCHE> — встановлення Android Studio, перший проект, GitHub*

*Опис PR (шаблон, скопіюйте й заповніть):*

*## Що зроблено*

*- Створено Android-проєкт \*\*CollegeHub-<PRIZVYSCHE>\*\* (Kotlin, Empty Activity)*

*- Додано Hello-екран із текстом "Hello, <PRIZVYSCHE>!"*

*- Створено власну папку: students/<PRIZVYSCHE>\_<IMYA>/lab1/*

*- Додано скріншот роботи: students/<PRIZVYSCHE>\_<IMYA>/lab1/docs/lab1-hello-<PRIZVYSCHE>.png*

*## Як перевірити*

*1. Відкрити проект: students/<PRIZVYSCHE>\_<IMYA>/lab1/CollegeHub-<PRIZVYSCHE>/*

*2. Зібрати `app` у Debug, запустити на AVD — має відобразитись текст "Hello, <PRIZVYSCHE>!"*

*## Чек-лист*

*- [ ] Проєкт збирається без помилок*

*- [ ] Є скріншот емулятора*

*- [ ] Шлях папок відповідає стандарту*

*- [ ] .gitignore присутній, зайві файли не завантажені*

*Додайте рев’юерів (викладача/асистентів), надішліть PR.*

***Увага: Лабораторна зараховується тільки через PR. Будь-які файли, прислані поза GitHub, не оцінюються.***

***9) Типові помилки та як їх виправити***

*git push просить токен: зайдіть у GitHub → Settings → Developer settings → Personal access tokens (Fine-grained), створіть токен і використайте його як пароль (або залогіньтесь через Android Studio).*

*«File name too long» у Windows: увімкніть git config --global core.longpaths true та переконайтесь, що шлях до проєкту не надто глибокий.*

*У PR з’явилися зайві файли (/build, .gradle): перевірте .gitignore, видаліть із гіта:*

*git rm -r --cached students/<PRIZVYSCHE>\_<IMYA>/lab1/CollegeHub-<PRIZVYSCHE>/.gradle*

*git rm -r --cached students/<PRIZVYSCHE>\_<IMYA>/lab1/CollegeHub-<PRIZVYSCHE>/app/build*

*git commit -m "chore(lab1): clean cached build artifacts"*

*git push*

***Додаткові матеріали***

1. *https://developer.android.com/studio/install*

***Зміст звіту***

1. *Мета роботи.*
2. *Короткий опис*
3. *Хід виконання роботи зі скріншотами*
4. *Аналіз результатів, висновки*