Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Лабораторна робота № 1

з курсу: «Розробка програмного забезпечення мобільних пристроїв»

За новим планом

**Виконав:**  
студент 4-го курсу,  
групи ТВ-23  
Губін Ярослав Олегович

Посилання на GitHub репозиторій: [https://github.com/YaroslavHubin](https://github.com/YaroslavHubin/Kotlin/tree/main/PW1%D0%A2%D0%92-23_%D0%93%D1%83%D0%B1%D1%96%D0%BD%D0%AF%D1%80%D0%BE%D1%81%D0%BB%D0%B0%D0%B2%D0%9E%D0%BB%D0%B5%D0%B3%D0%BE%D0%B2%D0%B8%D1%87)

**Перевірив:**

Недашківський О.Л.

Київ 2025/2026

Лабораторна робота №1

**Завдання:**

- ознайомлення з мовою Kotlin;

- встановлення середовища розробки;

- створення першого Android-проєкту у Android Studio з підтримкою Jetpack Compose для відображався привітального тексту з інтерфейсом введення імені (TextField) та кнопки, при натиску якої виводиться привітання з іменем;

- оголошення змінних, робота з типами та введенням з консолі;

- створення застосунку із введенням числового значення (TextField), кнопкою для підтвердження введення, реалізацією умовної логіки через if або when для відображення різних повідомлень/компонентів (наприклад відображати іконку в залежності від рівня заряду акумулятора);

- тестування застосунку та оформлення звіту зі створення першого

Android-проєкту.

**Хід виконання:**

Опис виконання завдань розділених на пункти аби спростити виконання лабораторної роботи на окремі під задачі для кращого розуміння:

- У завданні 1 було об’єднано встановлення середовища розробки та ознайомлення з мовою Kotlin.

- У завданні 2 описано створення першого Android-проєкту у Android Studio з підтримкою Jetpack Compose для відображався привітального тексту з інтерфейсом введення імені (TextField) та кнопки, при натиску якої виводиться привітання з іменем.

- У завданні 3 оголошення змінних, робота з типами та введенням з консолі.

- У завданні 4 створення застосунку із введенням числового значення (TextField), кнопкою для підтвердження введення, реалізацією умовної логіки через if або when для відображення різних повідомлень/компонентів.

- У завданні 5 – тестування застосунку та оформлення звіту зі створення першого Android-проєкту.

**Завдання 1:**

У якості середовищ розробки для мови Kotlin було використано Intellij IDEA та Android Studio. В Android Studio вже є можливість використання Jetpack Compose, як наведено на рисунку 1.

A screenshot of a computer

AI-generated content may be incorrect.

Рисунок 1 – Використання шаблону Jetpack Compose у Android Studio

А в Intellij IDEA менша кількість бібліотек, які можна одразу використовувати з Kotlin, проте можна легко писати код і одразу розібратись з веденням та виведенням з консолі, що можна використати для виконання завдання 3, де є потреба у використанні різних типів даних, оголошенням змінних і виведенням через консоль. На рисунку 2 наведено приклад виконання коду з лекції 4 в Intellij IDEA.

A screenshot of a computer

AI-generated content may be incorrect.

Рисунок 2 – Виконання прикладу коду з оголошенням змінної в Intellij IDEA

Як видно з наведених рисунків, то маленькі приклади на мові Kotlin працюють, що є фактом – правильного встановлення середовища розробки для мови програмування – Kotlin, а також базовим ознайомленням з самою мовою. Можна вважати цілі поставлені у завданні 1 – виконаними.

**Завдання 2:**

Для виконання цього завдання, необхідно зробити архітектуру з використанням Jetpack Compose. Я буду використовувати в роботі Android Studio, оскільки в ньому є вже одразу готовий шаблон для використання Jetpack Compose. Тому маємо таку створену вже архітектуру при обрані шаблону, як показано на рисунку 3.

A screenshot of a computer program

AI-generated content may be incorrect.

Рисунок 3 – Використання шаблону Jetpack Compose у Android Studio

На даний момент код не закоментований, оскільки він є шаблоном і його потрібно повністю переробити під нашу задачу. При додаванні певних функцій, які будуть потрібні для виконання тих чи інших завдань, коментування буде проходити у самому коді, тобто на скріншотах. Коментування при описі самих завдань буде мінімальним, аби зрозуміти лише базові дії.

Тому стерши майже весь код, переходимо до написання самої програми, у якій буде відображатись поле, де користувач зможе вести своє ім’я, а при натисканні кнопки, йому виведеться текстове повідомлення «Привіт, (Ведене користувачем ім’я)». Маємо наступний результат, зображений на рисунках 4-6.

A screenshot of a computer program

AI-generated content may be incorrect.

Рисунок 4 – Перша частина коду програми

A screenshot of a computer program

AI-generated content may be incorrect.

Рисунок 5 – Друга частина коду

A screenshot of a phone

AI-generated content may be incorrect.

Рисунок 6 – Результат виконання програми з привітанням користувача

**Завдання 3:**

Аби не розділяти проекти і не робити частину зі змінними та типами і виведенням і веденням з консолі, завдання буде зроблено у Android Studio, ніж те що зазначалось раніше про використання Intellij IDEA, Intellij IDEA можна використати, але для зручності і легкому перенесенні проекту на гіт хаб буде все зроблено в Android Studio. Тому натискаємо на File – New Module – Java or Kotlin Library і отримаємо такий результат у структурі проекту, як показано на рисунку 7.

A screenshot of a computer

AI-generated content may be incorrect.

Рисунок 7 – Створення окремого файлу для роботи з консоллю та типами даних

Тепер в цьому окремому файлі буде розміщено весь основний код, який стосується введення даних, їх виведення, типи та зміні. Більш детально розписано на рисунку 8.

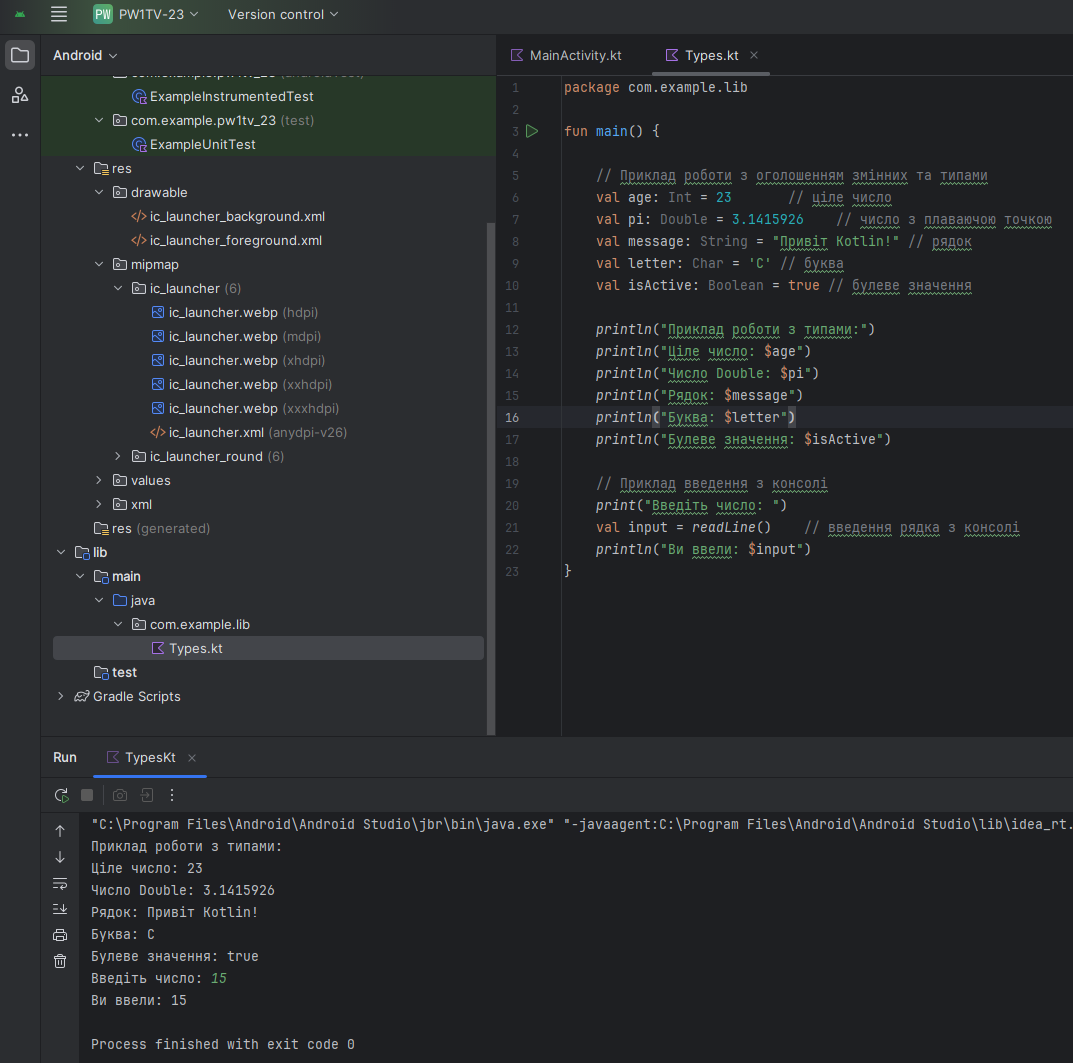


Рисунок 8 – Робота з типами даних, виведення та ведення

Окрім цього Kotlin дає можливість і інакше оголошувати змінні, зверніть увагу на рисунок 9 та 10.

A screenshot of a computer

AI-generated content may be incorrect.

Рисунок 9 – Приклад іншого задання змінних

A screen shot of a computer

AI-generated content may be incorrect.

Рисунок 10 – Приклад того, що змінні як вказано на рисунку 9 можна оголошувати

**Завдання 4:**

Зробивши окремий модуль для ведення та виведення різних типів даних та оголошень змінних, повернемось до основного проекту, де був результат виведення імені користувача по натисканню кнопки, і в цей же додаток додамо трохи більше функціоналу, згідно вимог 4-го завдання.

Для цього натиснувши ПКМ на поточний пакет, створюємо ще один котлін файл, який називаємо BatteryScreen, а в основний файл додаємо функцію виклику для BatteryScreen і коментуємо функцію для виклику додатку з привітанням користувача. Більш детальні зміни основного файлу наведені на рисунку 11.

A computer screen shot of a computer code

AI-generated content may be incorrect.

Рисунок 11 – Основні зміни у головному файлі

Тепер більш детально розглянемо файл, який буде відображати стан заряду акумулятора електромобіля. Даний файл можна викликати через основний – MainActivity. Детальний код та коментарі до коду наведені на рисунках 12-13.

A screenshot of a computer program

AI-generated content may be incorrect.

Рисунок 12 – Перша частина коду програми BatteryScreen

A screenshot of a computer program

AI-generated content may be incorrect.

Рисунок 13 – Друга частина коду програми BatteryScreen

Тепер розглянемо результат виконання даної програми і будемо вводити різні значення для перевірки умовної логіки оператора – if. Маємо результати зображені на рисунках 14 – 17.

A screenshot of a phone

AI-generated content may be incorrect.

Рисунок 14 – Результат для значення «1»

A screenshot of a phone

AI-generated content may be incorrect.

Рисунок 15 – Результат для значення «21»

A screenshot of a phone

AI-generated content may be incorrect.

Рисунок 16 – Результат для значення «51»

A screenshot of a phone

AI-generated content may be incorrect.

Рисунок 17 – Результат для значення «81»

Як видно з рисунків 14-17 умовна логіка працює справно і виводиться правильний результат користувачеві.

**Завдання 5:**

Також напишемо один юніт тест для тестування більш складної системи, а саме BatteryScreen. На інші файли писати юніт тести нема великої необхідності, оскільки вони були протестовані усі вручну, як видно з вище вказаних рисунків, а саме: 6, 8, 10, 14-17, хоча враховуючи що систему BatteryScreen було вже протестовано судячи з останніх скріншотів, я все ж таки вирішив написати маленькі юніт тести для цієї частини коду, і більш детально розглянемо цей код на рисунку 18.

A screenshot of a computer program

AI-generated content may be incorrect.

Рисунок 18 – Приклад unit тесту для BatteryScreen

Як видно з самого зеленого кольору та зеленої плашки навпроти тесту, то тест був успішний. Але задля цікавості змінемо значення у рядку 19. І маємо такий результат на рисунку 19.

A screenshot of a computer

AI-generated content may be incorrect.

Рисунок 19 – Спеціально змінимо значення в unit тесті, аби видало помилку

Як видно з рисунку 19, тест при вказані замалого значення, яке не входить під параметри видає помилку, що свідчить про те що на рисунку 18 був правильно прописаний юніт тест і що сам додаток BatteryScreen працює коректно.

**Висновок:**

Під час виконання лабораторної роботи 1 було встановлено середовище розробки – Android Studio та Intellij IDEA для роботи з такою мовою програмування, як Kotlin. Під час виконання практичної роботи було розроблену два мобільні додатки, які вдало запускаються на емуляторі мобільного телефону, а також правильно працюють і відповідають зазначеним у завданнях вимогах. Окрім цього, було створено один консольний додаток, аби пропрацювати різне оголошення змінних, та їх типів та спробувати ввести та вивести значення у консолі. Всі ці три файли є частиною одного проекту на мові програмування Kotlin. Під час виконання лабораторної роботи було освоєно базові навички роботи з цією мовою програмування, а також було написано один unit тест для 1 – єї з програм. Всі поставленні завдання під час виконання лабораторної роботи були виконані. Код був достатньою мірою закоментований.