Mobile Programming - Assignment 1

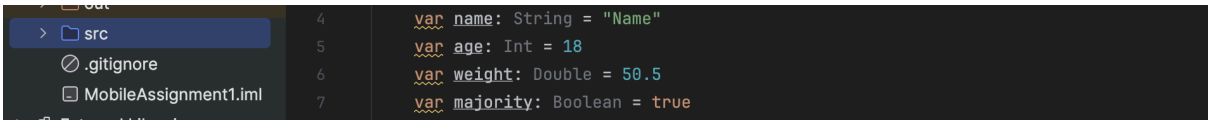
First and Last Name: Tamiris Abildayeva

Link to GitHub:
https://github.com/TamirisK/university-mobile-programming/tree/main/%5BMobile%5D%20Assignment%201%20by%20Abildayeva%20T

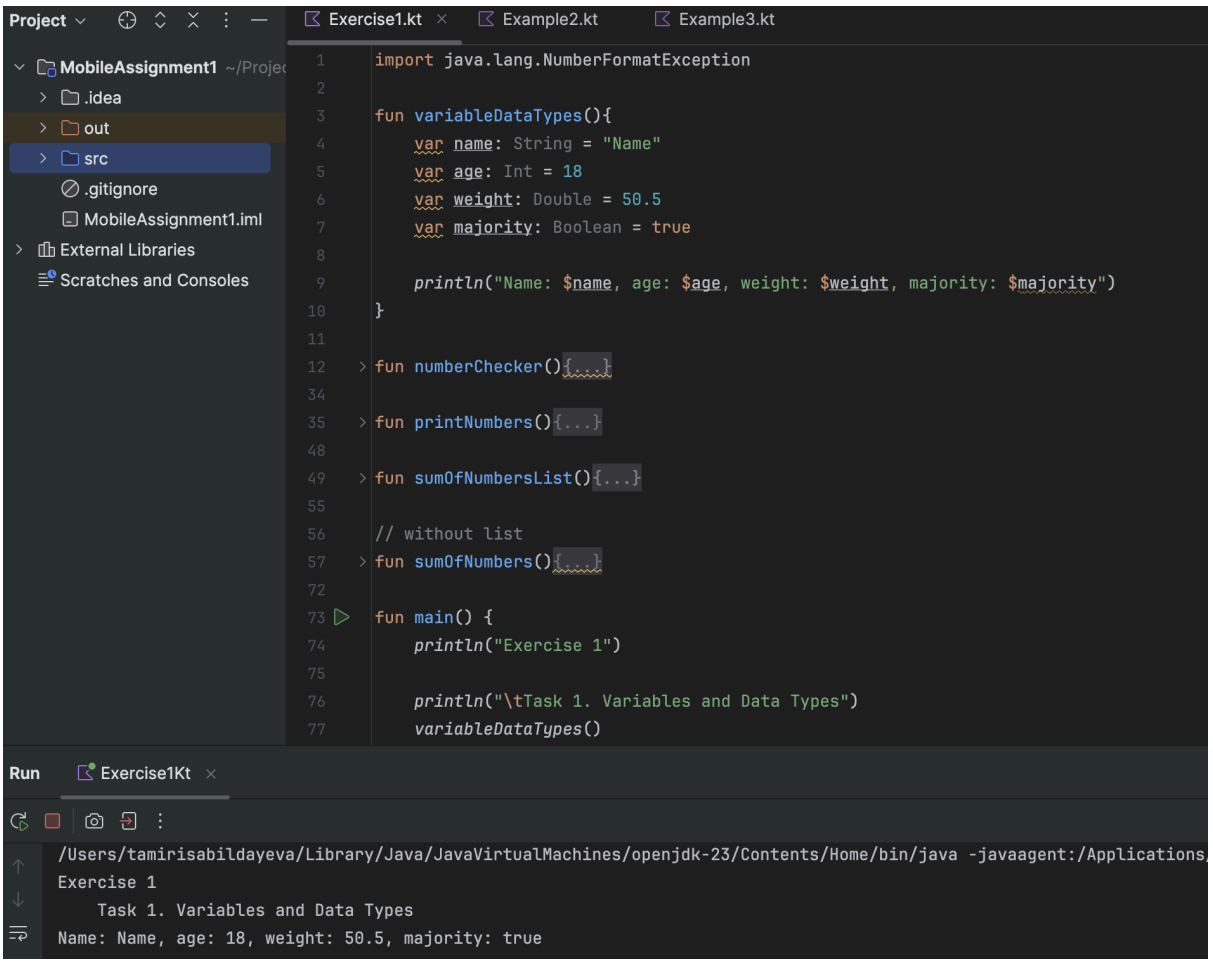
https://drive.google.com/drive/folders/1ZblzGDlgmGuCa04rBePHQNr_lc21Gu9n?usp=share_link

Exercise 1: Kotlin Syntax Basics

1. **Variables and Data Types:**
   - Create variables of different data types: `Int`, `Double`, `String`, `Boolean`.



   - Print the variables using `println`.



Conditional Statements:

- Create a simple program that checks if a number is positive, negative, or zero.

```kotlin
fun numberChecker(){
    while (true){
        println("Enter any number. To exit enter any word")

        try {
            var number:Int = readln().toInt()

            if (number > 0) println("Number $number is positive")
            if (number == 0) println("Number $number is zero")
            if (number < 0) println("Number $number is negative")

//            Another version of if case
//            number > 0 -> println("Number $number is positive")
//            number == 0 -> println("Number $number is zero")
//            number < 0 -> println("Number $number is negative")

        } catch (e: NumberFormatException){
            println("Not number entered \nGood Bye \nExiting...")
            break
        }
```

```
/Users/tamirisabildayeva/Library/Java/JavaVirtualMachines/openjdk-23/Contents/Home/bin/java -javaagent:/Applications/
Exercise 1

    Task 2. Conditional Statements
Enter any number. To exit enter any word
1
Number 1 is positive
Enter any number. To exit enter any word
-1
Number -1 is negative
Enter any number. To exit enter any word
0
Number 0 is zero
```

Loops:

- Write a program that prints numbers from 1 to 10 using `for` and `while` loops

```
MobileAssignment1.iml
External Libraries
Scratches and Consoles
```

```kotlin
fun printNumbers(){
    print("With For Loop: ")
    for (i in 1 ≤ .. ≤ 10){
        print("$i ")
    }

    print("\nWith While Loop: ")
    var i: Int = 1
    while (i <= 10){
        print("$i ")
        i++
    }
}
```

Run Exercise1Kt

```
/Users/tamirisabildayeva/Library/Java/JavaVirtualMachines/openjdk-23/Contents/Home/bin/java -javaagent:/Applications/
Exercise 1

    Task 3. Loops
With For Loop: 1 2 3 4 5 6 7 8 9 10
With While Loop: 1 2 3 4 5 6 7 8 9 10
```

Collections:

- Create a list of numbers, iterate through the list, and print the sum of all numbers.

```kotlin
fun sumOfNumbersList(){
    val numbers = listOf(1, 2, 3, 4, 5)
    val sum = numbers.sum()
    println("Numbers of the list is: $numbers")
    println("The sum of the list is: $sum")
}

// without list
fun sumOfNumbers(){
    var sum: Int = 0

    while (true){
        println("Enter any number. To exit enter any word")

        try {
            var number:Int = readln().toInt()
            sum += number
        } catch (e: NumberFormatException){
            println("Calculating..\nSum: $sum")
            break
```

```
/Users/tamirisabildayeva/Library/Java/JavaVirtualMachines/openjdk-23/Contents/Home/bin/java -javaagent:/Applications
Exercise 1
    Task 4. Collections
Numbers of the list is: [1, 2, 3, 4, 5]
The sum of the list is: 15
Enter any number. To exit enter any word
1
Enter any number. To exit enter any word
2
Enter any number. To exit enter any word
exit
Calculating..
Sum: 3
```

Exercise 2: Kotlin OOP (Object-Oriented Programming)

1. **Create a `Person` class:**
    - Define properties for `name`, `age`, and `email`.

```kotlin
open class Person(
    var name: String,
    var age: Int,
    var email: String
) {
```

- Create a method to display the person's details.

```kotlin
open class Person(
    var name: String,
    var age: Int,
    var email: String
) {

//    override fun toString(): String{
    open fun displayInfo(){
        print("Name: $name, Age: $age, Email: $email")
    }
}
```

```
/Users/tamirisabildayeva/Library/Java/JavaVirtualMachines/openjdk-23/Contents/Home/bin/java -javaagent:/Applications
Exercise 2
    Task 1. Create a Person class
Name: Tamiris, Age: 20, Email: tamiris@example.com
```

Inheritance:

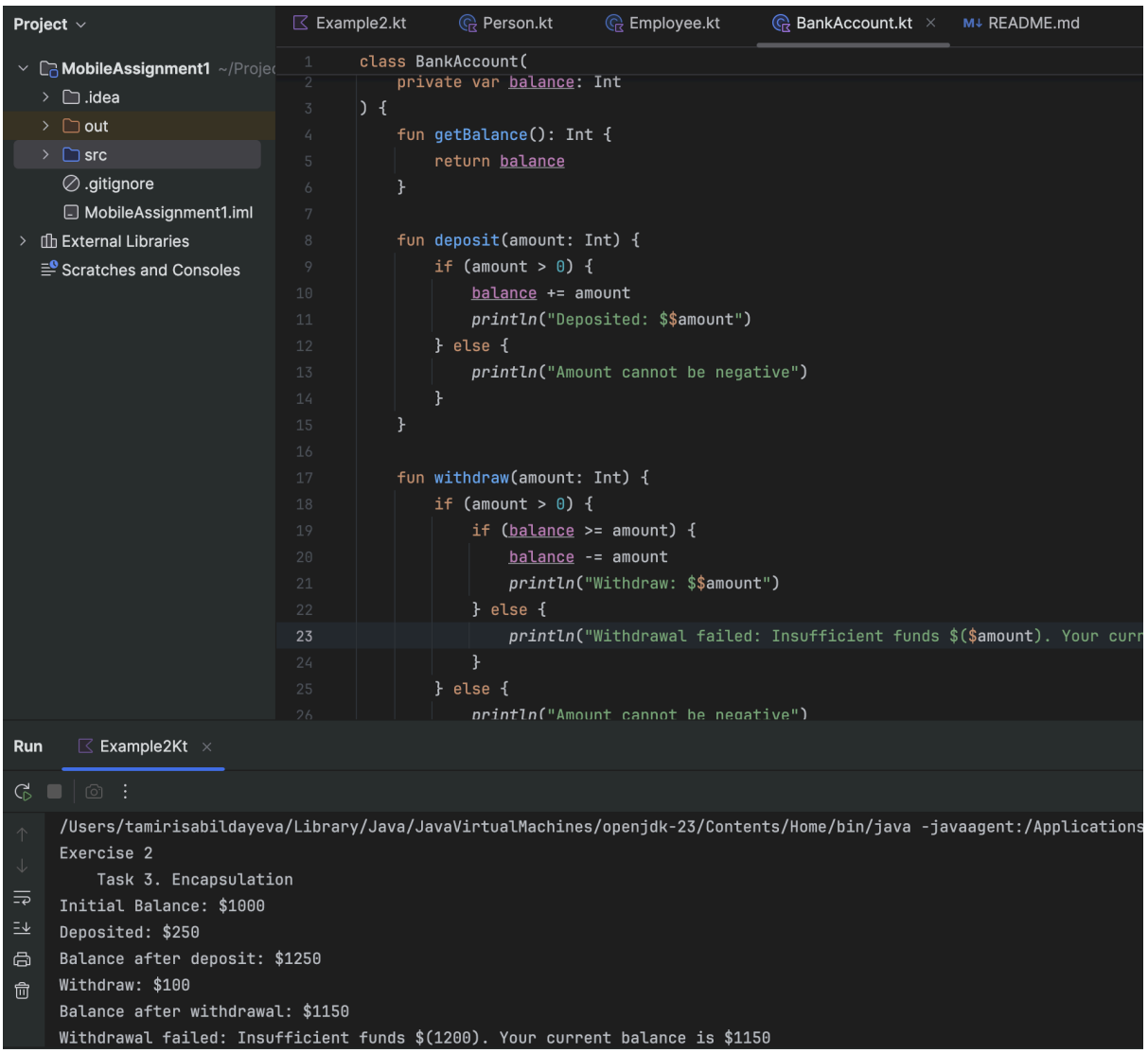
- Create a class `Employee` that inherits from the `Person` class.
- Add a property for `salary`.
- Override the `displayInfo` method to include the salary.

```kotlin
class Employee(
    name: String,
    age: Int,
    email: String,

    var salary: Int
) : Person (name, age, email){

    override fun displayInfo(){
        super.displayInfo()
        println(", Salary: $salary")
//        return "Name: $name, Age: $age, Email: $email, Salary: $salary"
    }
}
```

```
/Users/tamirisabildayeva/Library/Java/JavaVirtualMachines/openjdk-23/Contents/Home/bin/java -javaagent:/Applications
Exercise 2
\tTask 2. Inheritance
Name: Tomas, Age: 25, Email: tomas@example.com, Salary: 1000
```

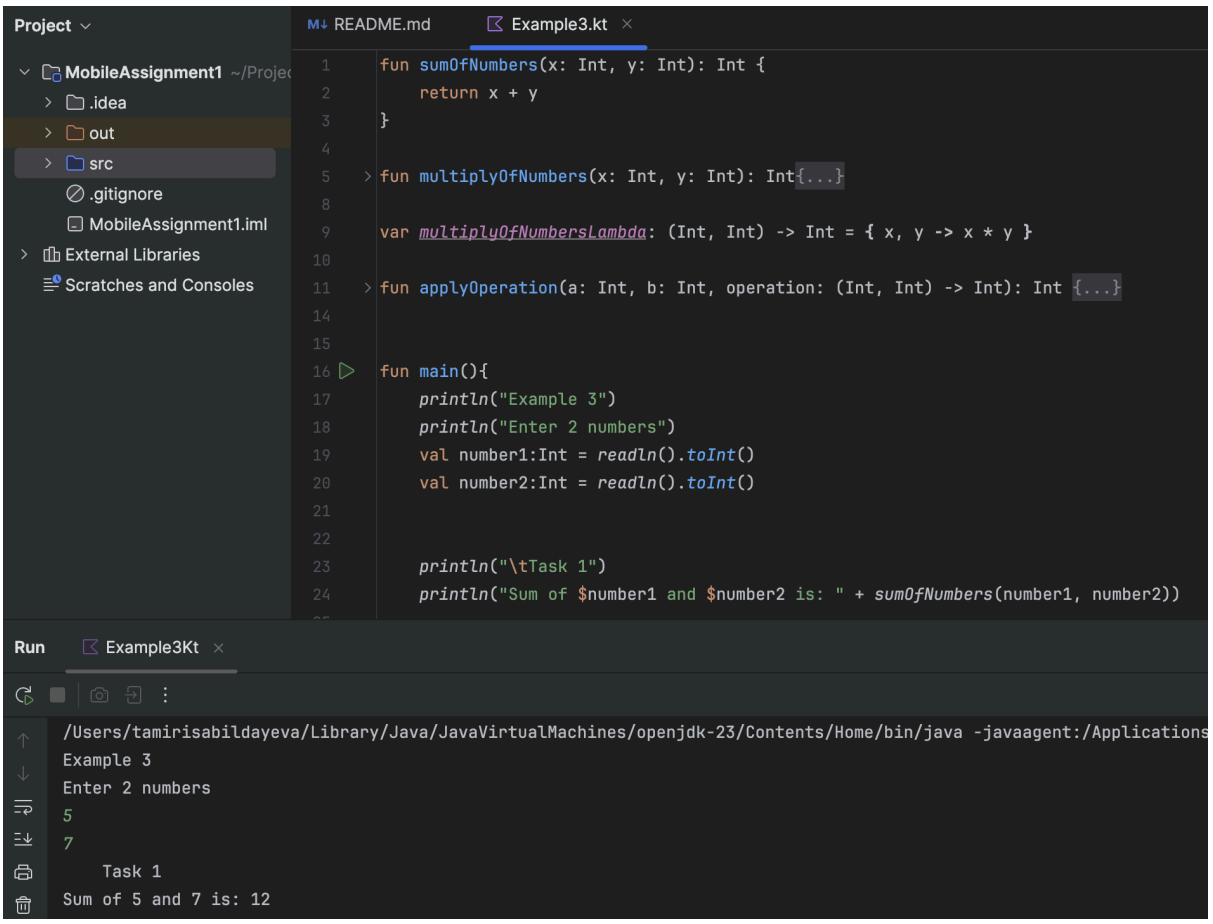
Encapsulation:

- Create a `BankAccount` class with a private property `balance`.
- Provide methods to `deposit` and `withdraw` money, ensuring the balance never goes negative.



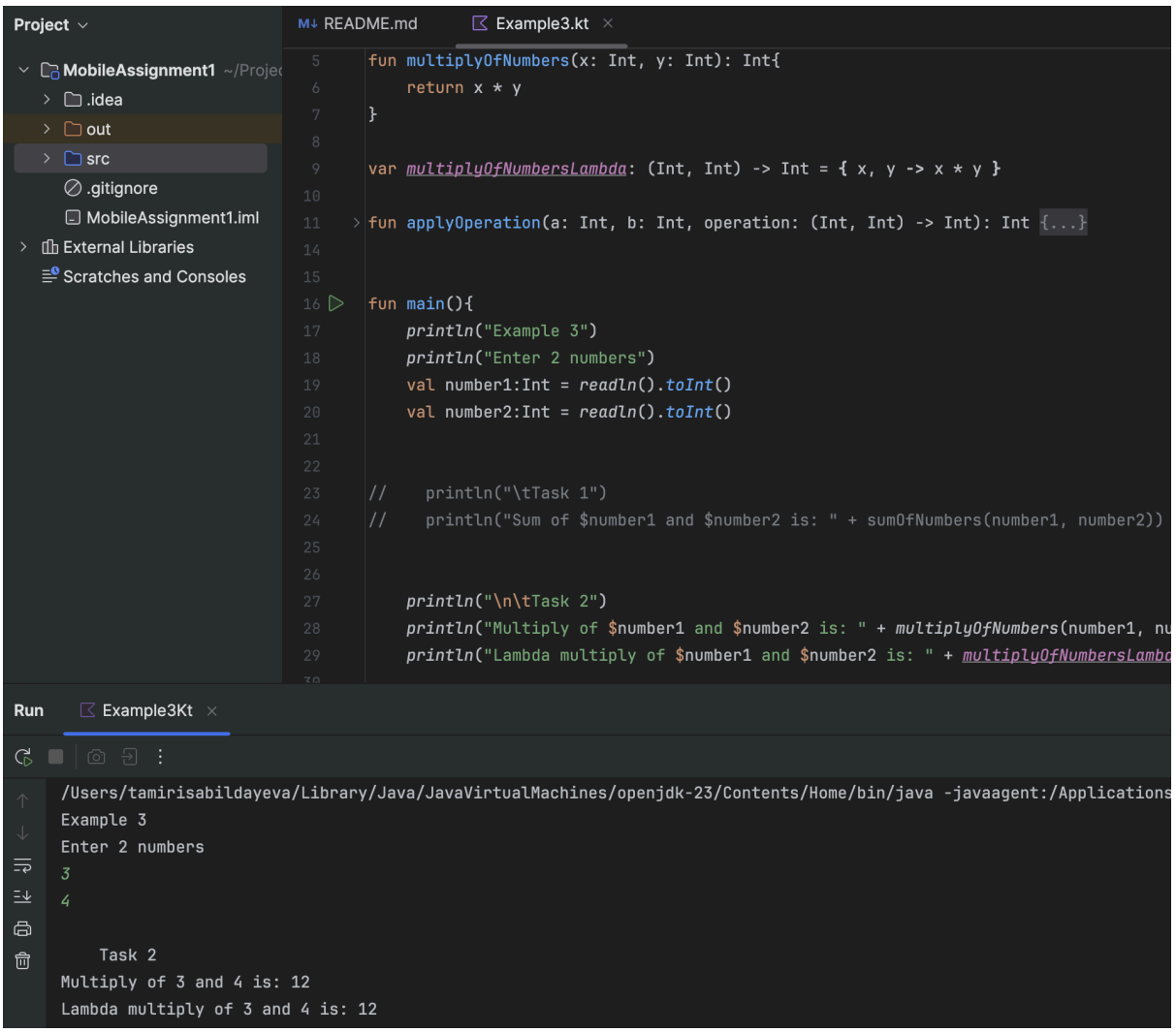
Exercise 3: Kotlin Functions

1. **Basic Function:**
   - Write a function that takes two integers as arguments and returns their sum



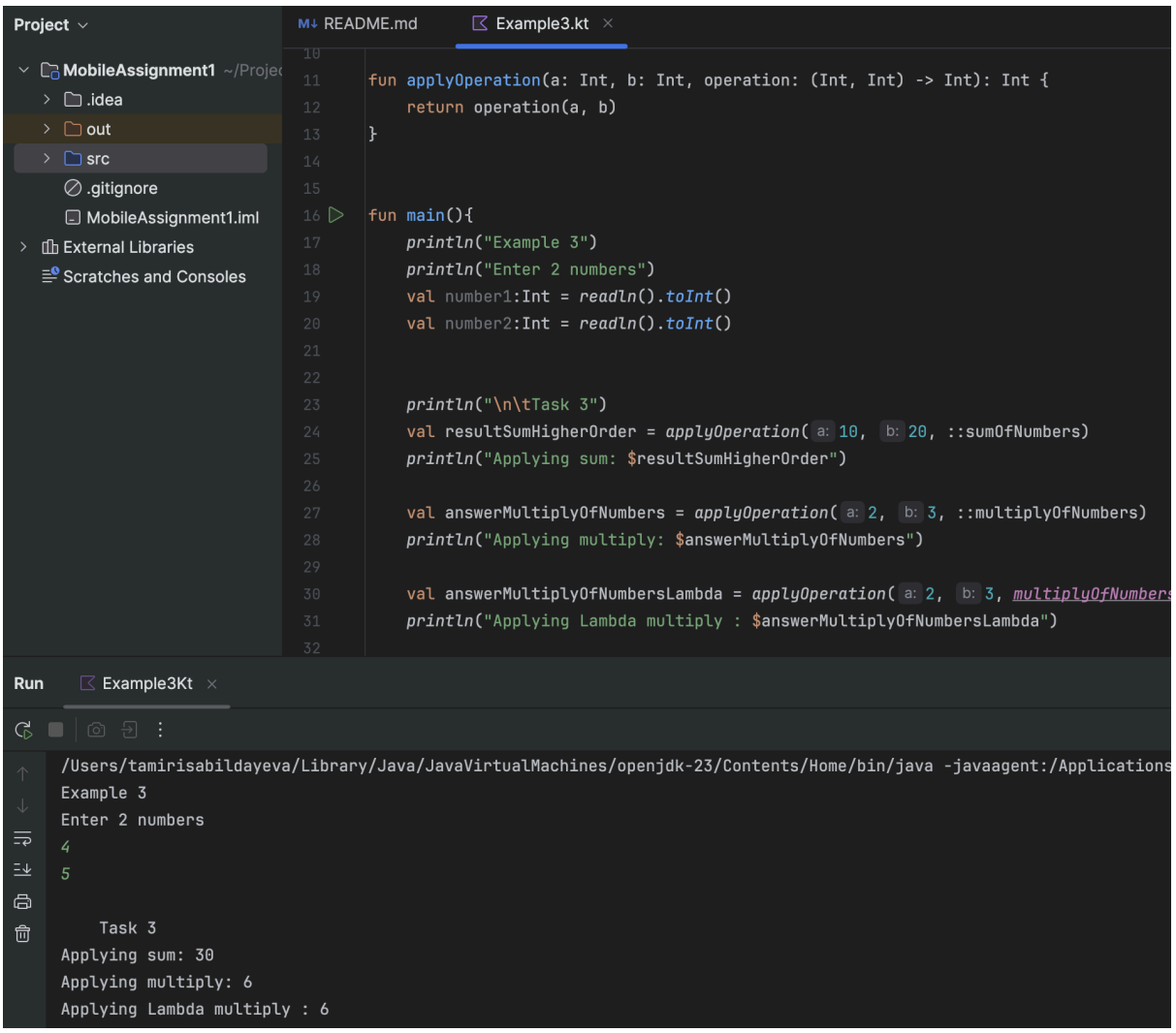
Lambda Functions:

- Create a lambda function that multiplies two numbers and returns the result



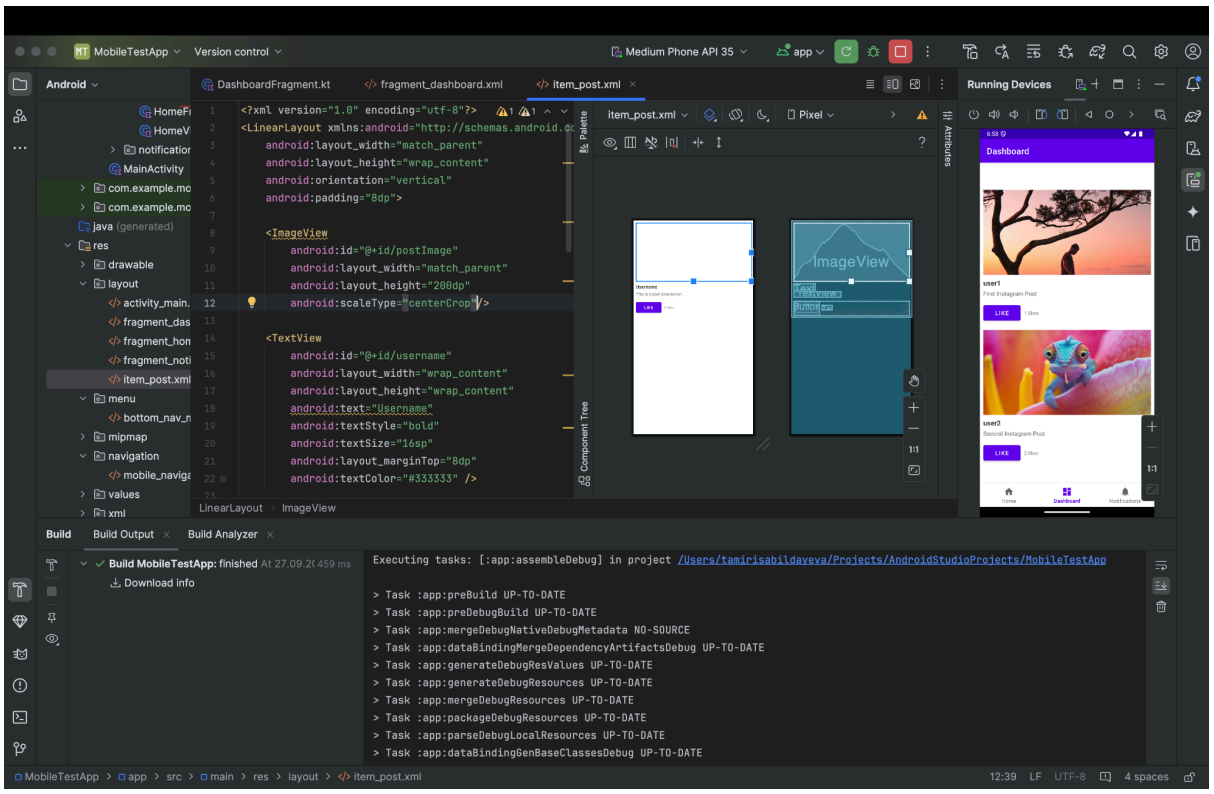
Higher-Order Functions:

- Write a function that takes a lambda function as a parameter and applies it to two integers.



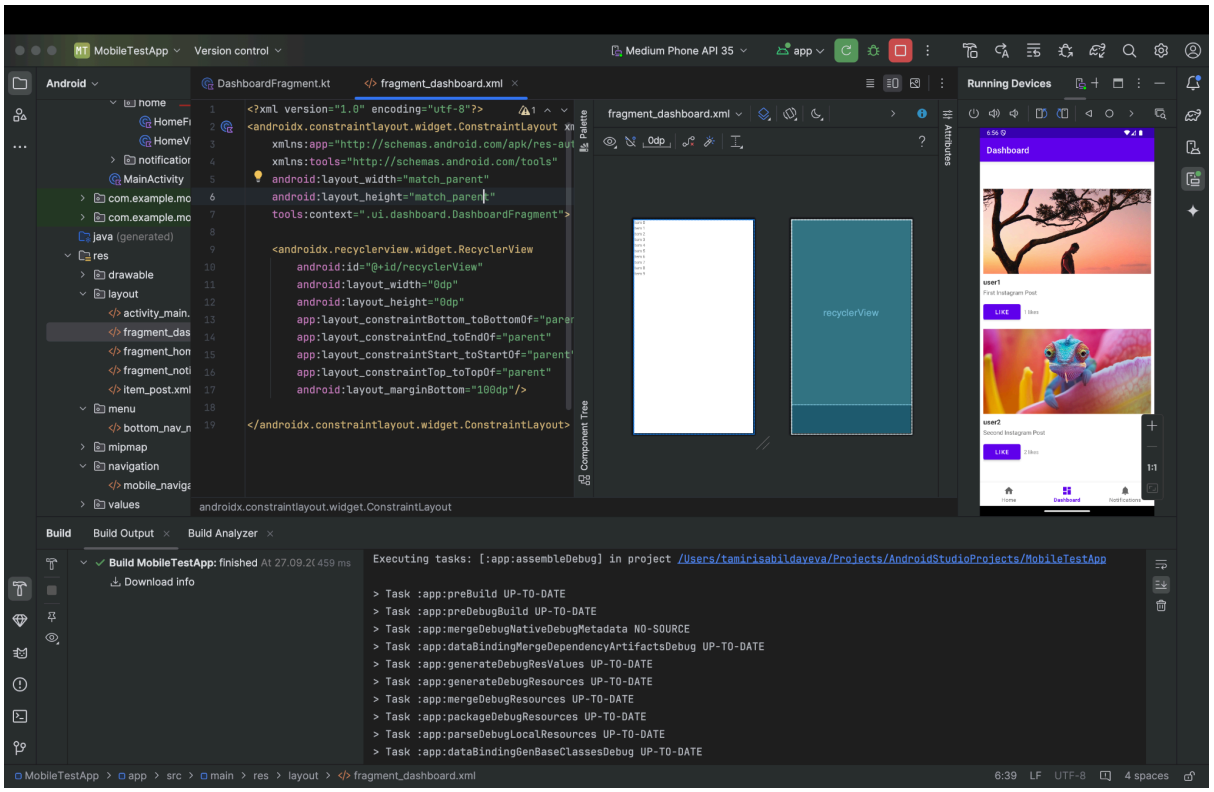
Exercise 4: Android Layout in Kotlin (Instagram-like Layout)

1. **Set Up the Android Project:**
   - Create a new Android project in Android Studio.
   - Ensure you have a Kotlin-based project.
2. **Design the Layout:**
   - Create a new XML layout file (`activity_main.xml`) for a simple Instagram-like user interface.
   - Include elements like `ImageView`, `TextView`, and `RecyclerView` for the feed



Create the RecyclerView Adapter:

- Set up the RecyclerView to display a feed of posts with `ImageView` for the picture and `TextView` for the caption.



MainActivity Setup:

- Initialize the `RecyclerView` in `MainActivity` and populate it with sample data