Assignment 1, Mobile Programming

Makhatbek Iliyas

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.

Loops:

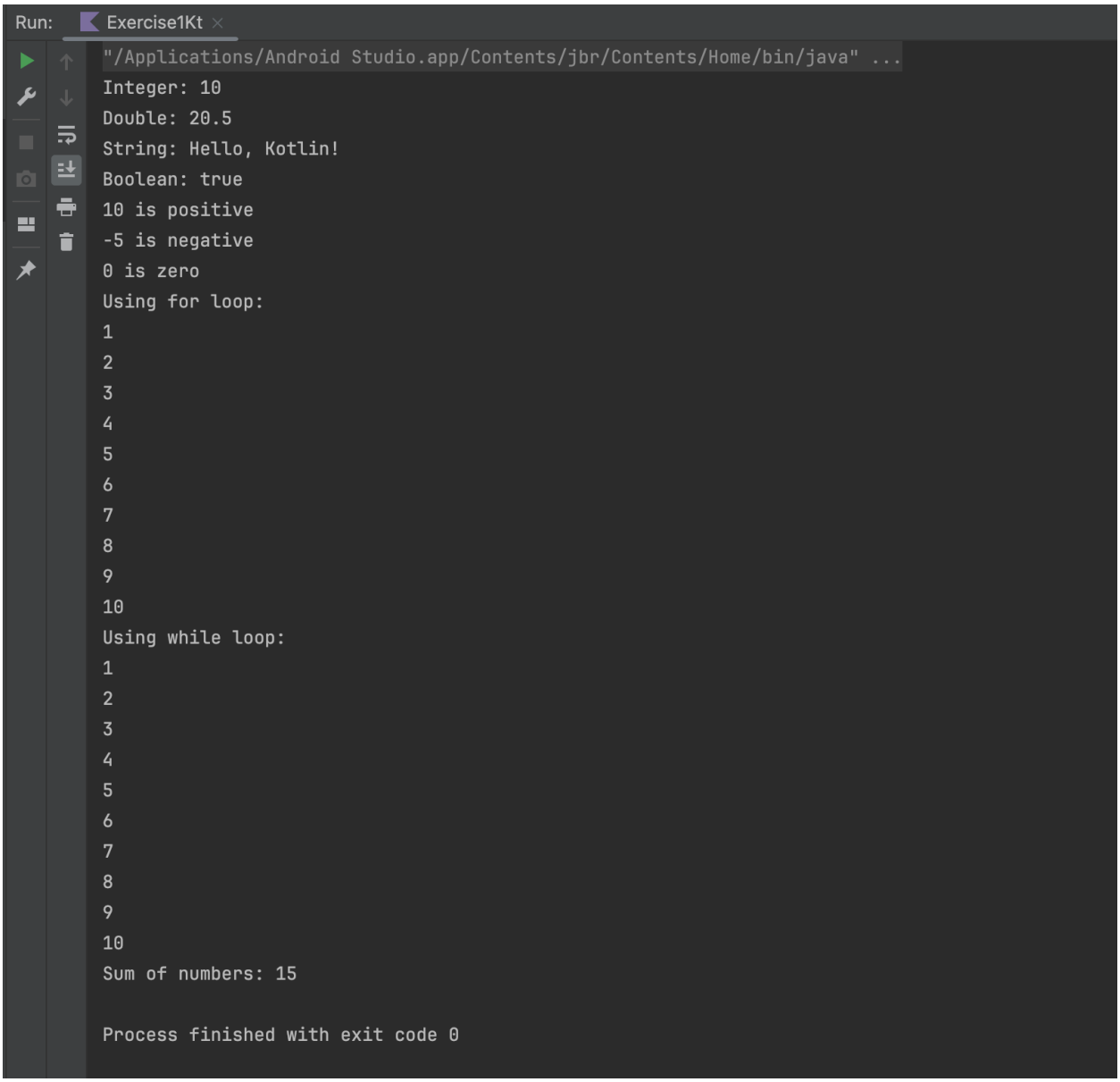
- Write a program that prints numbers from 1 to 10 using `for` and `while` loops

Collections:

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

```kotlin
package com.example.kotlinbasics

// 1. Variables and Data Types
fun main() {
    // Declare variables of different data types
    val myInt: Int = 10
    val myDouble: Double = 20.5
    val myString: String = "Hello, Kotlin!"
    val myBoolean: Boolean = true

    println("Integer: $myInt")
    println("Double: $myDouble")
    println("String: $myString")
    println("Boolean: $myBoolean")

    // 2. Conditional Statements
    checkNumber( num: 10)
    checkNumber( num: -5)
    checkNumber( num: 0)

    // 3. Loops
    printNumbers()

    // 4. Collections
    sumOfList()
}

// Conditional Statements
fun checkNumber(num: Int) {
    if (num > 0) {
        println("$num is positive")
    } else if (num < 0) {
        println("$num is negative")
    } else {
        println("$num is zero")
    }
```

```kotlin
}

// Loops
fun printNumbers() {
    println("Using for loop:")
    for (i in 1 ≤ .. ≤ 10) {
        println(i)
    }

    println("Using while loop:")
    var i = 1
    while (i <= 10) {
        println(i)
        i++
    }
}

// Collections
fun sumOfList() {
    val numbers = listOf(1, 2, 3, 4, 5)
    var sum = 0

    for (num in numbers) {
        sum += num
    }

    println("Sum of numbers: $sum")
}

```



Exercise 2: Kotlin OOP (Object-Oriented Programming)

1. **Create a `Person` class:**
    - Define properties for `name`, `age`, and `email`.
    - Create a method to display the person's details.

Inheritance:

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

Encapsulation:

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

```kotlin
package com.example.kotlinbasics

open class Person(private val name: String, private val age: Int, private val email: String) {
    open fun displayInfo() {
        println("Name: $name, Age: $age, Email: $email")
    }
}

class Employee(name: String, age: Int, email: String, private val salary: Double) : Person(name, age, email) {
    override fun displayInfo() {
        super.displayInfo()
        println("Salary: $salary")
    }
}

class BankAccount {
    private var balance: Double = 0.0

    fun deposit(amount: Double) {
        if (amount > 0) {
            balance += amount
            println("Deposited: $amount, New Balance: $balance")
        } else {
            println("Invalid deposit amount")
        }
    }

    fun withdraw(amount: Double) {
        if (amount > 0 && amount <= balance) {
            balance -= amount
            println("Withdrew: $amount, New Balance: $balance")
        } else {
            println("Insufficient balance or invalid amount")
        }
    }
```

```kotlin

    fun getBalance(): Double {
        return balance
    }
}

fun main() {
    // Test Person Class
    val person = Person( name: "Derbis Erzhan",  age: 22,  email: "yerslan@gmail.com")
    person.displayInfo()

    // Test Employee Class
    val employee = Employee( name: "Sabit Dara",  age: 28,  email: "dara@gmail.com",  salary: 60000.0)
    employee.displayInfo()

    // Test BankAccount Class
    val bankAccount = BankAccount()
    bankAccount.deposit( amount: 1000.0)
    bankAccount.withdraw( amount: 500.0)
    bankAccount.withdraw( amount: 600.0)
    println("Final Balance: ${bankAccount.getBalance()}")
}

```

```
Exercise2Kt
"/Applications/Android Studio.app/Contents/jbr/Contents/Home/bin/java" ...
Name: Derbis Erzhan, Age: 22, Email: yerslan@gmail.com
Name: Sabit Dara, Age: 28, Email: dara@gmail.com
Salary: 60000.0
Deposited: 1000.0, New Balance: 1000.0
Withdrew: 500.0, New Balance: 500.0
Insufficient balance or invalid amount
Final Balance: 500.0

Process finished with exit code 0
```

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.

```kotlin
package com.example.kotlinbasics

fun sum(a: Int, b: Int): Int {
    return a + b
}


val multiply: (Int, Int) -> Int = { x, y -> x * y }


fun operate(a: Int, b: Int, operation: (Int, Int) -> Int): Int {
    return operation(a, b)
}

fun main() {

    val result = sum( a: 10,  b: 20)
    println("Sum: $result")


    val product = multiply(5, 4)
    println("Product: $product")


    val additionResult = operate( a: 10,  b: 5, ::sum)
    println("Addition using Higher-Order Function: $additionResult")

    val multiplicationResult = operate( a: 6,  b: 7, multiply)
    println("Multiplication using Higher-Order Function: $multiplicationResult")
}

```

```
Exercise3Kt
"/Applications/Android Studio.app/Contents/jbr/Contents/Home/bin/java" ...
Sum: 30
Product: 20
Addition using Higher-Order Function: 15
Multiplication using Higher-Order Function: 42

Process finished with exit code 0
```

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

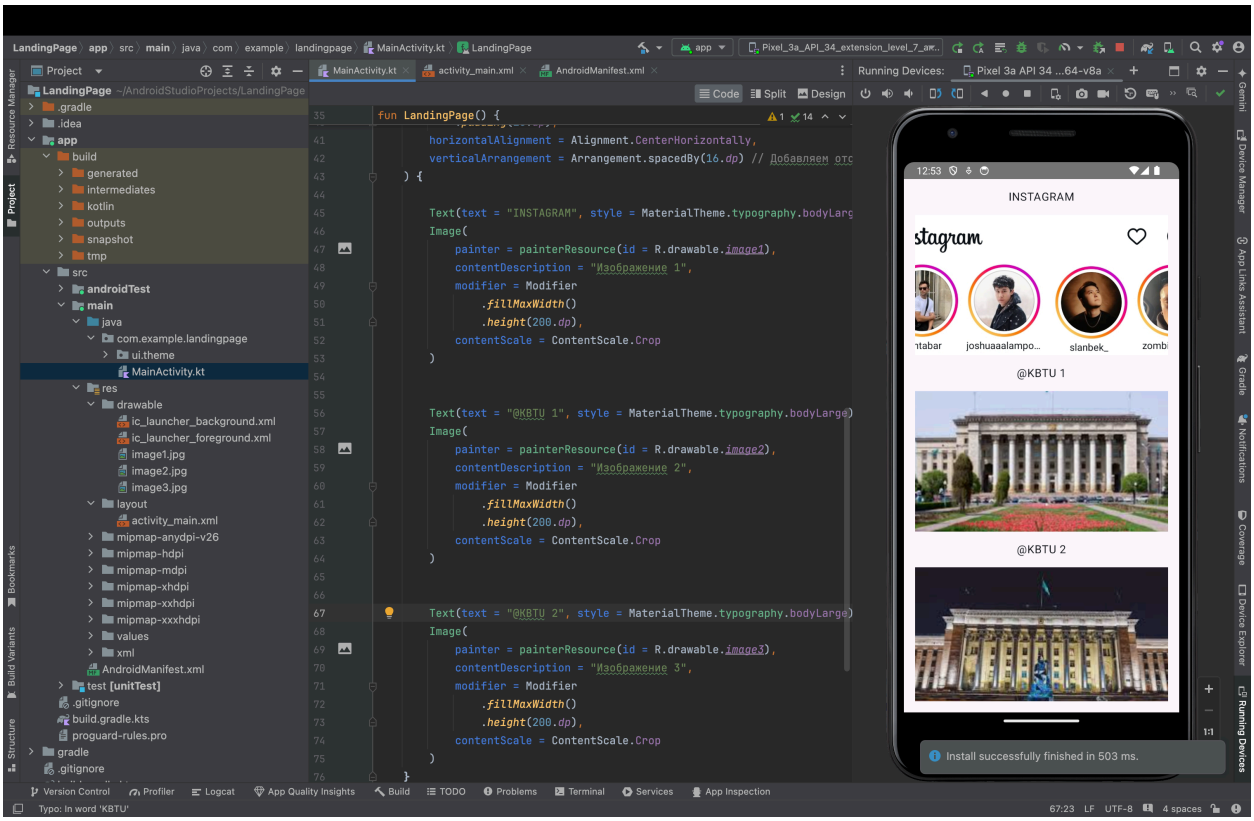
1. **Set Up the Android Project:**
    - Create a new Android project in Android Studio.
    - Ensure you have a Kotlin-based project.
2. **Design the Layout:**
    - Create a new XML layout file (`activity_main.xml`) for a simple Instagram-like user interface.
    - Include elements like `ImageView`,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

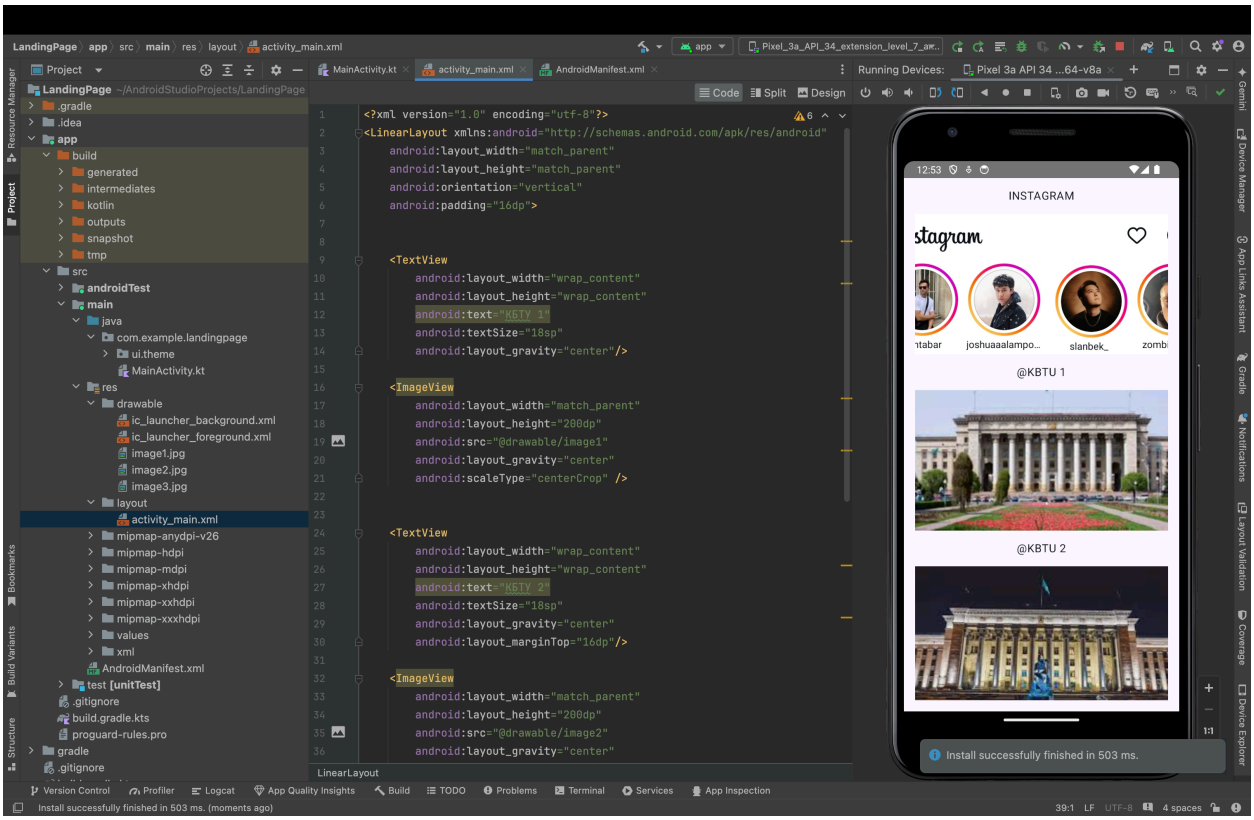
Create the RecyclerView Adapter:

- 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




LandingPage app src main res layout activity_main.xml
Project
LandingPage ~/AndroidStudioProjects/LandingPage
.gradle
.idea
app
build
generated
intermediates
kotlin
outputs
snapshot
tmp
src
androidTest
main
java
com.example.landingpage
ui.theme
MainActivity.kt
res
drawable
ic_launcher_background.xml
ic_launcher_foreground.xml
image1.jpg
image2.jpg
image3.jpg
layout
activity_main.xml
mipmap-anydpi-v26
mipmap-hdpi
mipmap-mdpi
mipmap-xhdpi
mipmap-xxhdpi
mipmap-xxxhdpi
values
xml
AndroidManifest.xml
test [unitTest]
.gitignore
build.gradle.kts
proguard-rules.pro
gradle
.gitignore
MainActivity.kt
activity_main.xml
AndroidManifest.xml
Code
Split
Design
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:padding="16dp">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="KBTU 1"
android:textSize="18sp"
android:layout_gravity="center"/>
<ImageView
android:layout_width="match_parent"
android:layout_height="200dp"
android:src="@drawable/image1"
android:layout_gravity="center"
android:scaleType="centerCrop" />
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="KBTU 2"
android:textSize="18sp"
android:layout_gravity="center"
android:layout_marginTop="16dp"/>
<ImageView
android:layout_width="match_parent"
android:layout_height="200dp"
android:src="@drawable/image2"
android:layout_gravity="center"
LinearLayout
Running Devices:
Pixel 3a API 34 ...64-v8a
12:53
INSTAGRAM
stagram
@KBTU 1
@KBTU 2
Install successfully finished in 503 ms.
Version Control
Profiler
Logcat
App Quality Insights
Build
TODO
Problems
Terminal
Services
App Inspection
Install successfully finished in 503 ms. (moments ago)
39:1
LF
UTF-8
4 spaces