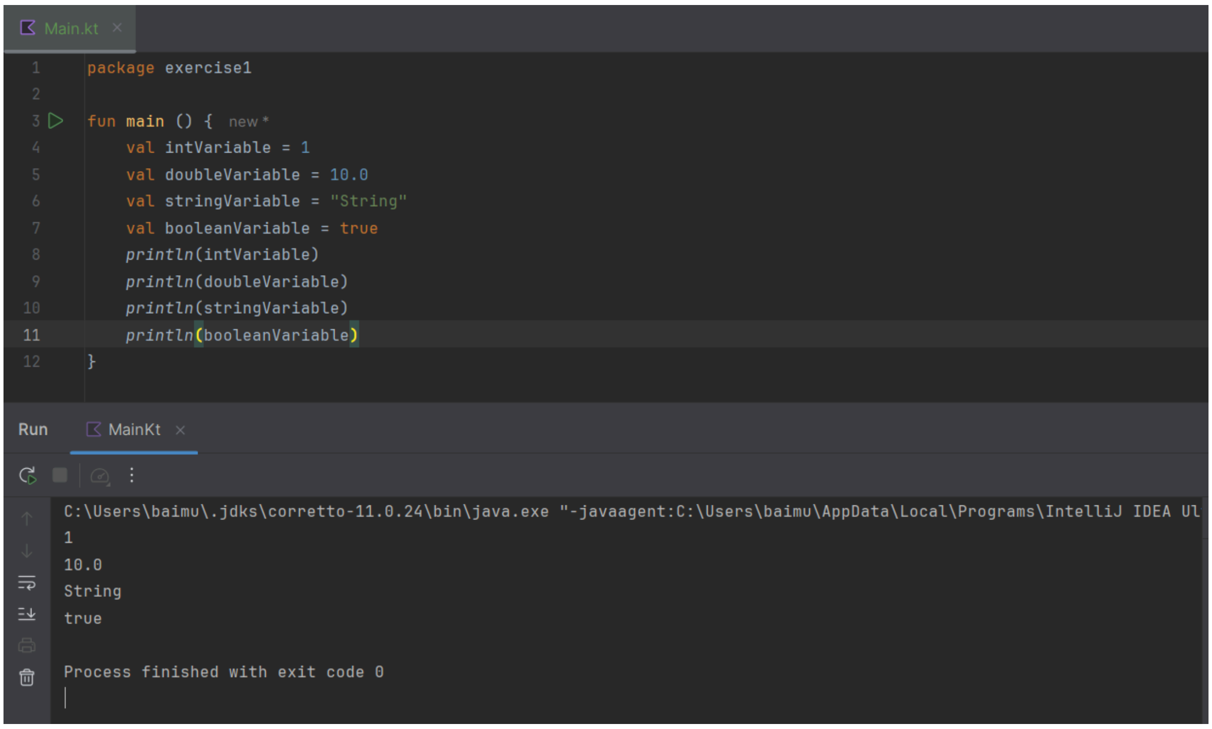
Assignment 1, Mobile Programming

Rafael Baimurzin

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`.

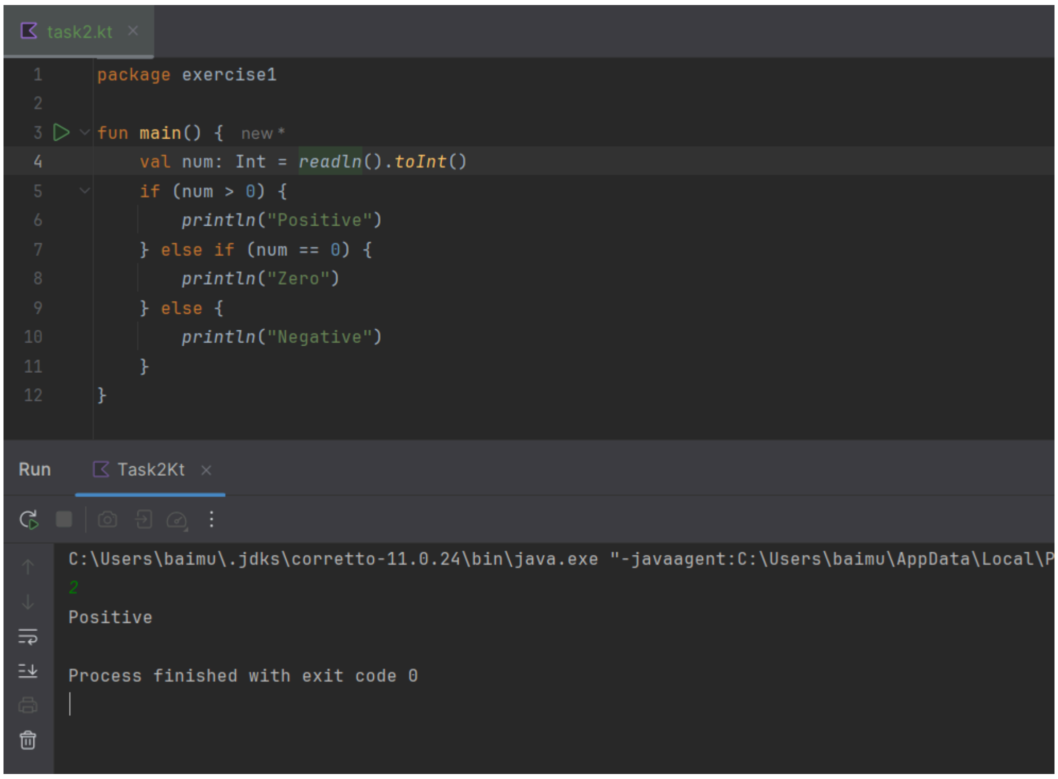


```kotlin
package exercise1

fun main () {
    val intVariable = 1
    val doubleVariable = 10.0
    val stringVariable = "String"
    val booleanVariable = true
    println(intVariable)
    println(doubleVariable)
    println(stringVariable)
    println(booleanVariable)
}
```

```
C:\Users\baimu\.jdks\corretto-11.0.24\bin\java.exe "-javaagent:C:\Users\baimu\AppData\Local\Programs\IntelliJ IDEA Ul
1
10.0
String
true

Process finished with exit code 0
```

Conditional Statements:

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

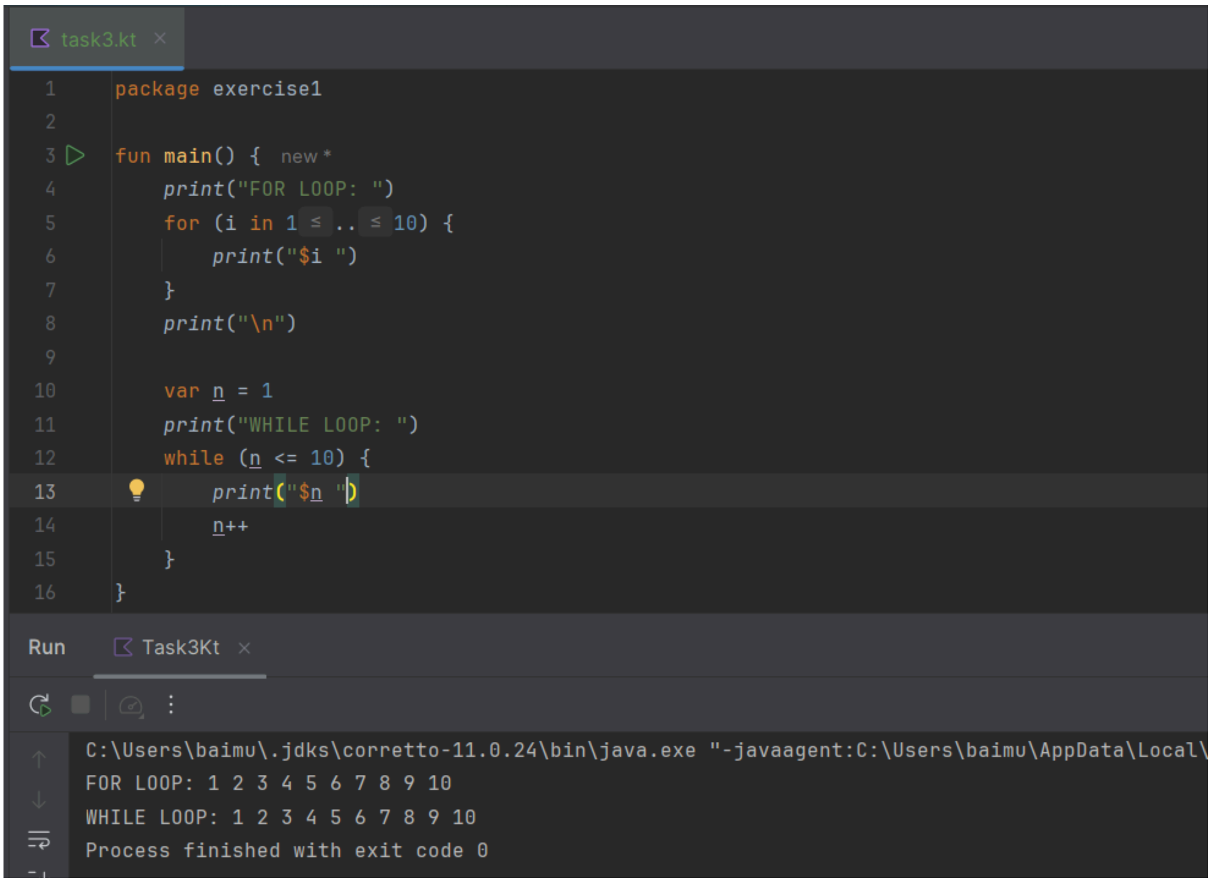
- 

```kotlin
package exercise1

fun main() {
    val num: Int = readln().toInt()
    if (num > 0) {
        println("Positive")
    } else if (num == 0) {
        println("Zero")
    } else {
        println("Negative")
    }
}
```

```
C:\Users\baimu\.jdks\corretto-11.0.24\bin\java.exe "-javaagent:C:\Users\baimu\AppData\Local\P
2
Positive

Process finished with exit code 0
```

Loops:

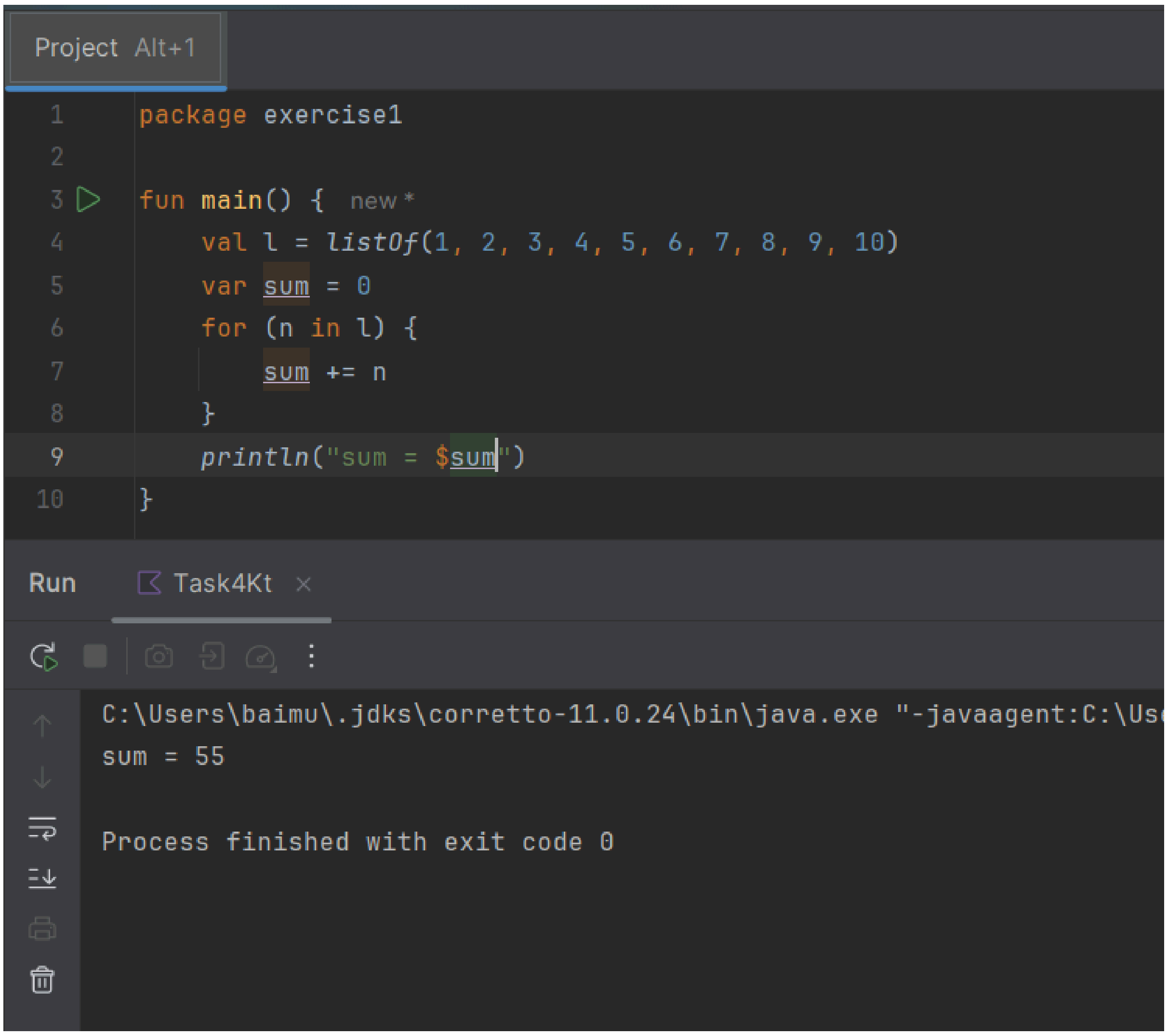
- Write a program that prints numbers from 1 to 10 using `for` and `while` loops

- 

```kotlin
package exercise1

fun main() {
    print("FOR LOOP: ")
    for (i in 1 ≤ .. ≤ 10) {
        print("$i ")
    }
    print("\n")

    var n = 1
    print("WHILE LOOP: ")
    while (n <= 10) {
        print("$n ")
        n++
    }
}
```

```
C:\Users\baimu\.jdks\corretto-11.0.24\bin\java.exe "-javaagent:C:\Users\baimu\AppData\Local\
FOR LOOP: 1 2 3 4 5 6 7 8 9 10
WHILE LOOP: 1 2 3 4 5 6 7 8 9 10
Process finished with exit code 0
```

Collections:

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

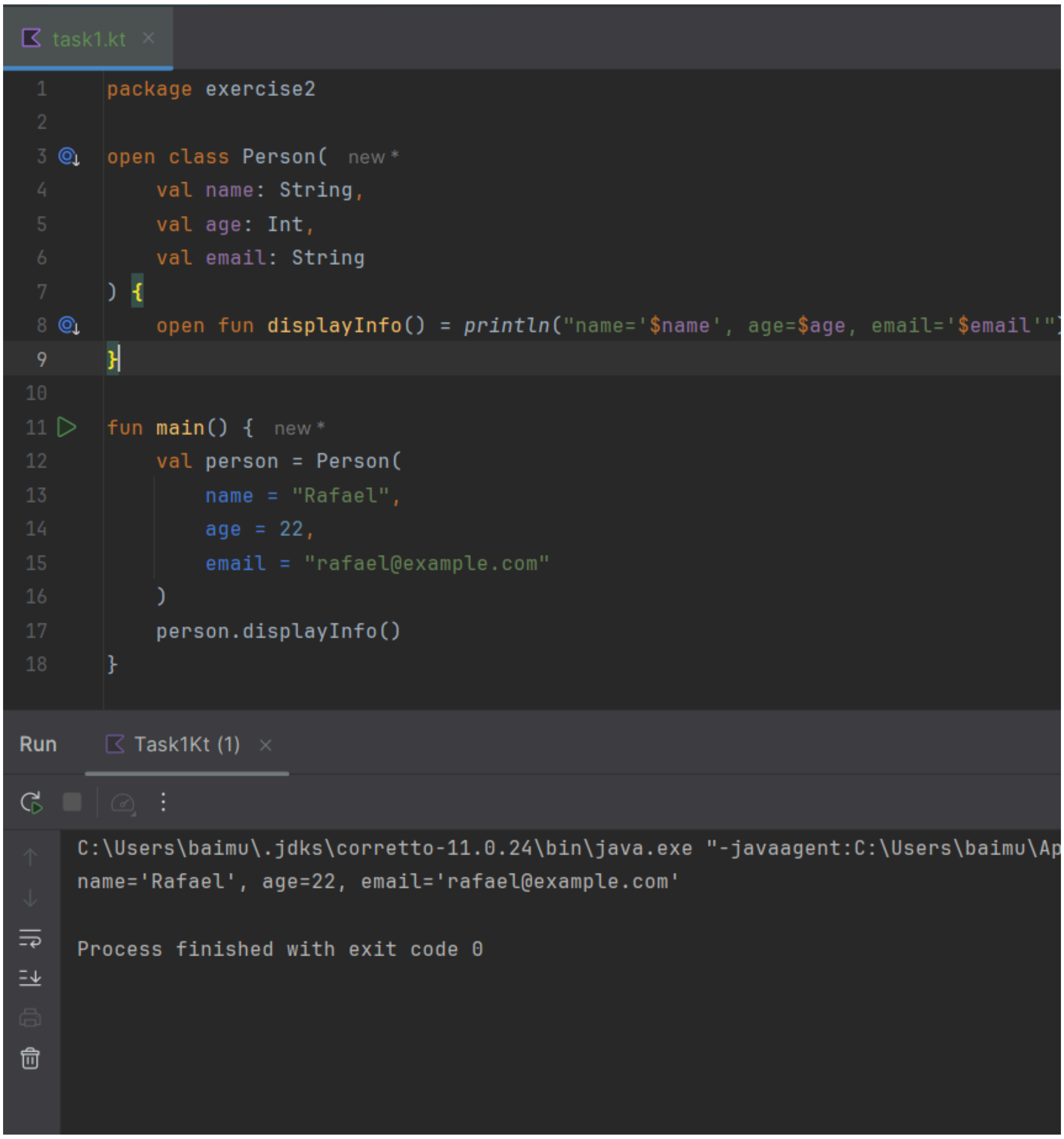
```kotlin
package exercise1

fun main() {  new *
    val l = listOf(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
    var sum = 0
    for (n in l) {
        sum += n
    }
    println("sum = $sum")
}
```

```
C:\Users\baimu\.jdks\corretto-11.0.24\bin\java.exe "-javaagent:C:\Us
sum = 55

Process finished with exit code 0
```

Exercise 2: Kotlin OOP (Object-Oriented Programming)

Create a `Person` class:

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

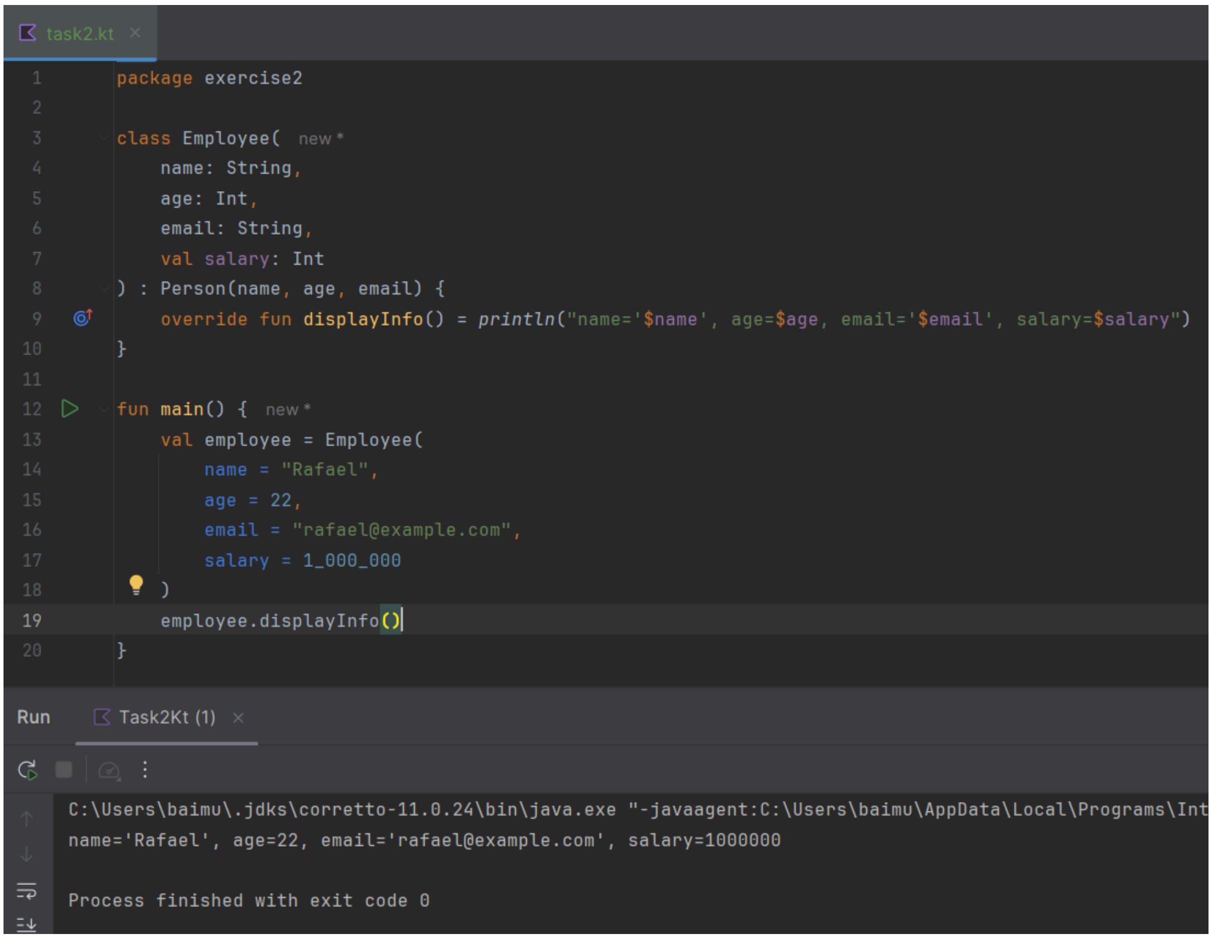


```kotlin
package exercise2

open class Person(  new *
    val name: String,
    val age: Int,
    val email: String
) {
    open fun displayInfo() = println("name='$name', age=$age, email='$email'"
}

fun main() {  new *
    val person = Person(
        name = "Rafael",
        age = 22,
        email = "rafael@example.com"
    )
    person.displayInfo()
}
```

```
C:\Users\baimu\.jdks\corretto-11.0.24\bin\java.exe "-javaagent:C:\Users\baimu\Ap
name='Rafael', age=22, email='rafael@example.com'

Process finished with exit code 0
```

Inheritance:

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

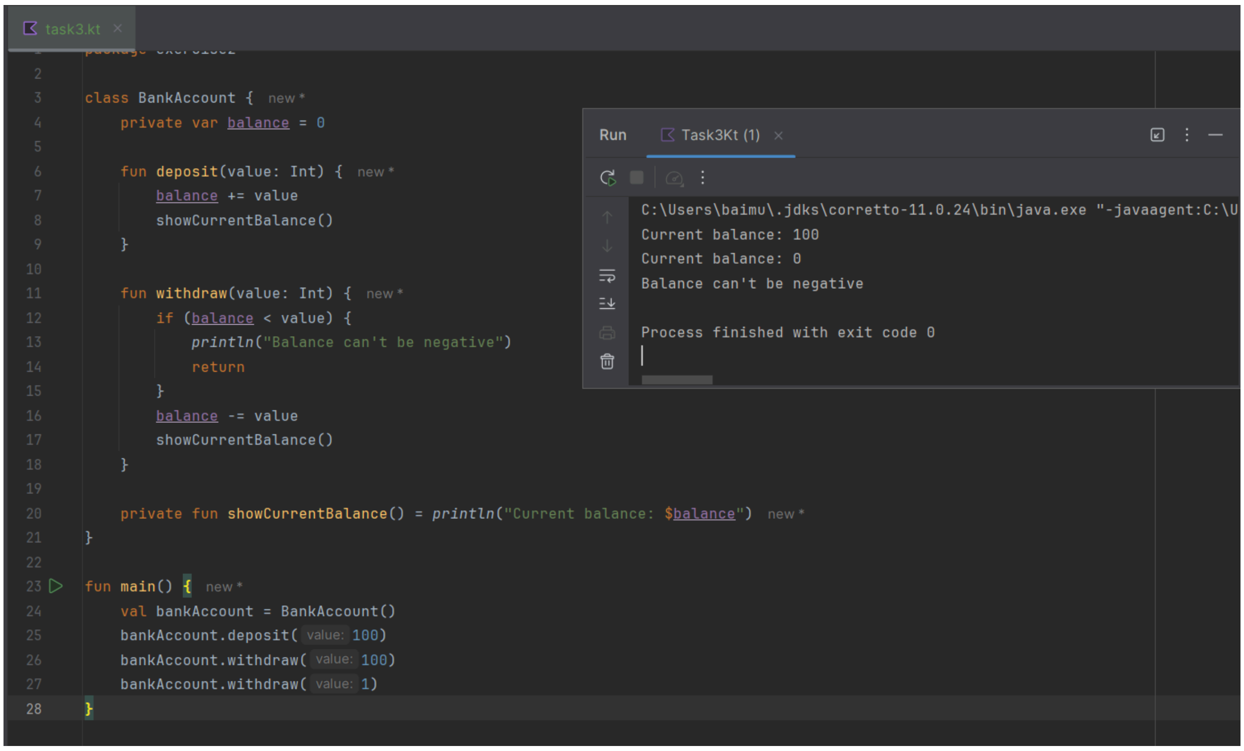


```kotlin
package exercise2

class Employee(  new *
    name: String,
    age: Int,
    email: String,
    val salary: Int
) : Person(name, age, email) {
    override fun displayInfo() = println("name='$name', age=$age, email='$email', salary=$salary")
}

fun main() {  new *
    val employee = Employee(
        name = "Rafael",
        age = 22,
        email = "rafael@example.com",
        salary = 1_000_000
    )
    employee.displayInfo()
}
```

```
C:\Users\baimu\.jdks\corretto-11.0.24\bin\java.exe "-javaagent:C:\Users\baimu\AppData\Local\Programs\Int
name='Rafael', age=22, email='rafael@example.com', salary=1000000

Process finished with exit code 0
```

Encapsulation:

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

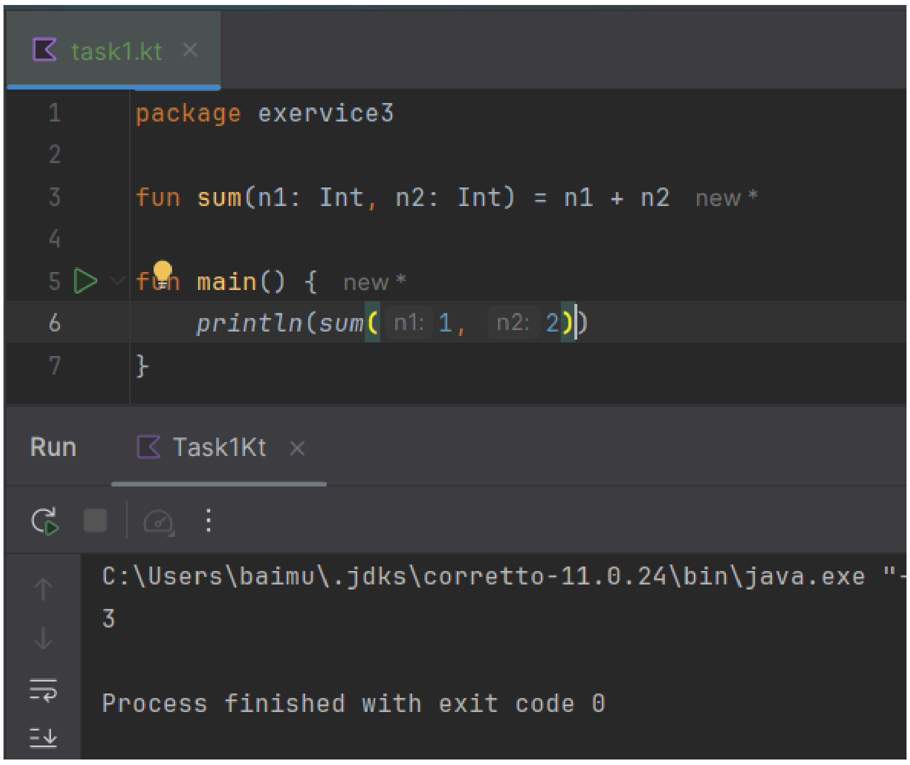


```kotlin
package exercise2

class BankAccount {
    private var balance = 0

    fun deposit(value: Int) {
        balance += value
        showCurrentBalance()
    }

    fun withdraw(value: Int) {
        if (balance < value) {
            println("Balance can't be negative")
            return
        }
        balance -= value
        showCurrentBalance()
    }

    private fun showCurrentBalance() = println("Current balance: $balance")
}

fun main() {
    val bankAccount = BankAccount()
    bankAccount.deposit(100)
    bankAccount.withdraw(100)
    bankAccount.withdraw(1)
}
```

```
C:\Users\baimu\.jdks\corretto-11.0.24\bin\java.exe "-javaagent:C:\U
Current balance: 100
Current balance: 0
Balance can't be negative

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

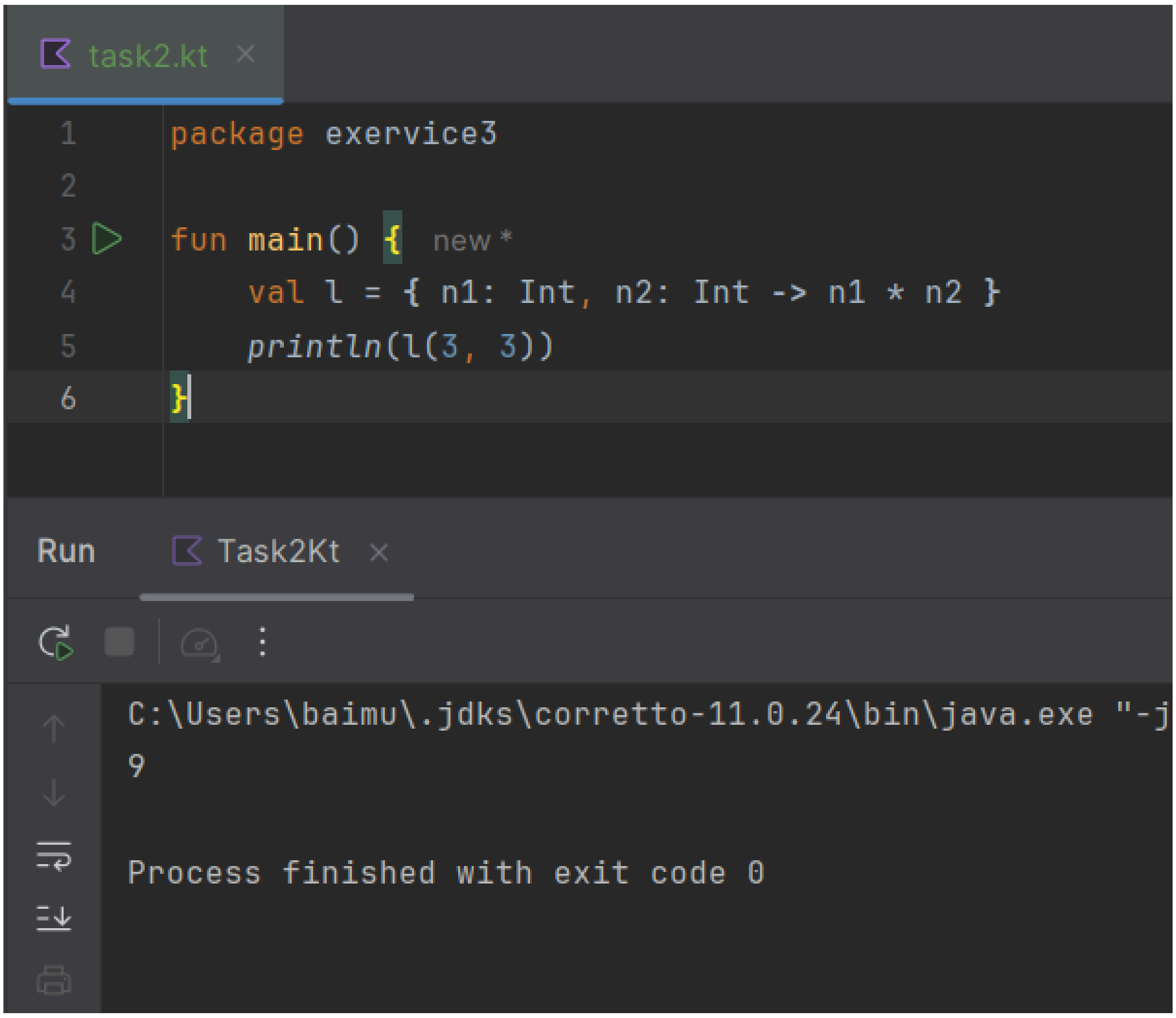


```kotlin
package exervice3

fun sum(n1: Int, n2: Int) = n1 + n2

fun main() {
    println(sum(1, 2))
}
```

```
C:\Users\baimu\.jdks\corretto-11.0.24\bin\java.exe "-
3

Process finished with exit code 0
```

Lambda Functions:

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

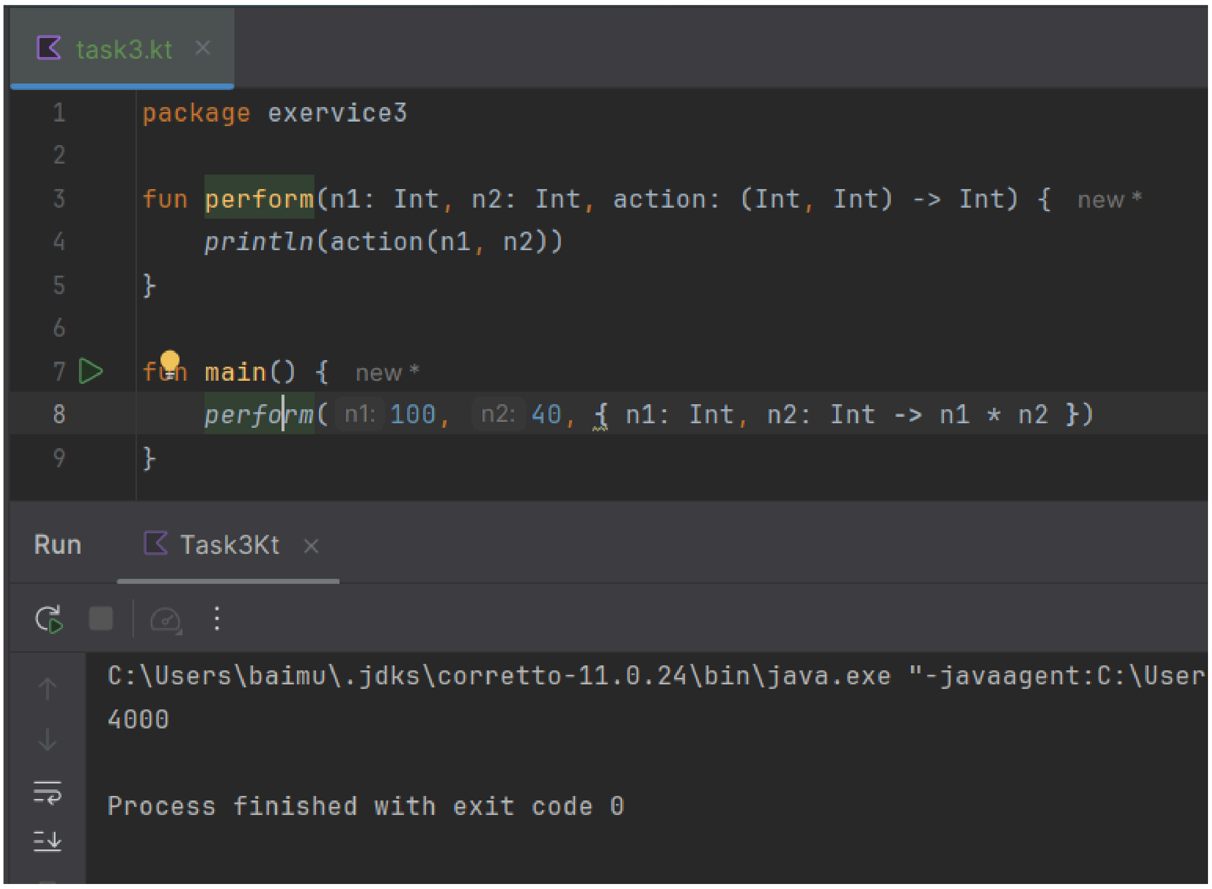


```kotlin
package exervice3

fun main() {
    val l = { n1: Int, n2: Int -> n1 * n2 }
    println(l(3, 3))
}
```

```
C:\Users\baimu\.jdks\corretto-11.0.24\bin\java.exe "-j
9

Process finished with exit code 0
```

Higher-Order Functions:

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


task3.kt

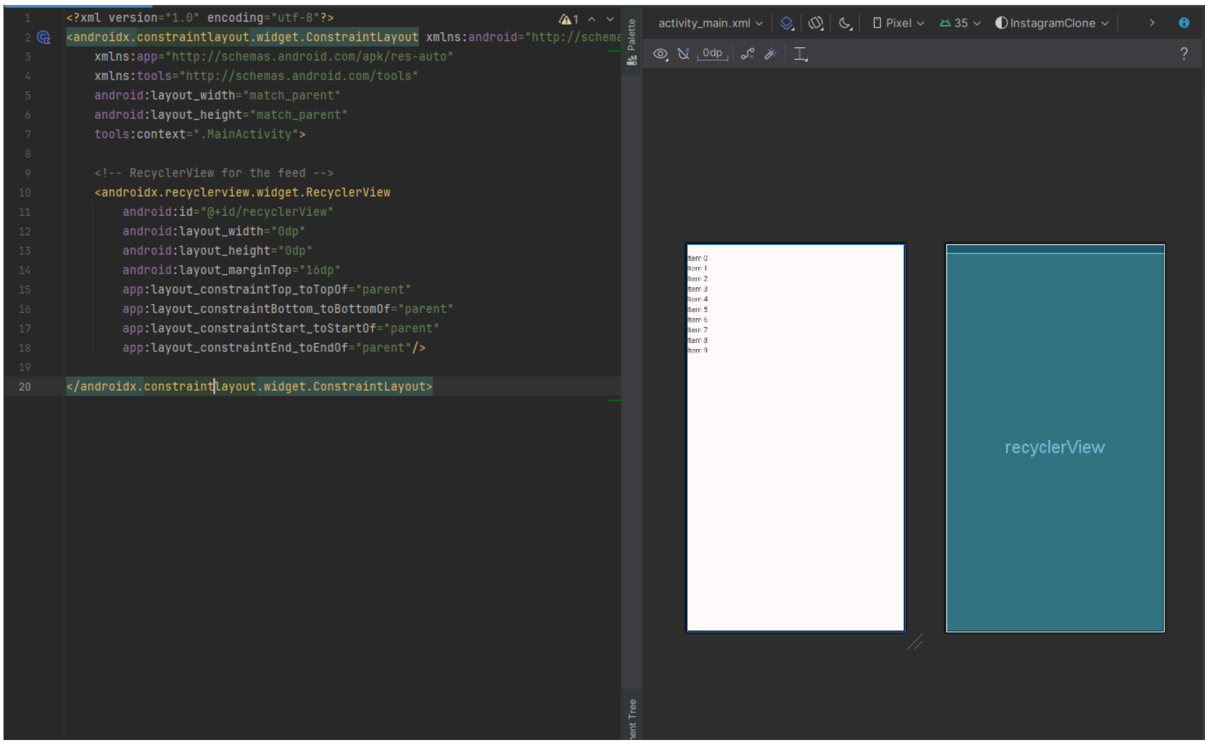
```
package exervice3

fun perform(n1: Int, n2: Int, action: (Int, Int) -> Int) {  new *
    println(action(n1, n2))
}

fun main() {  new *
    perform( n1: 100,  n2: 40, { n1: Int, n2: Int -> n1 * n2 })
}
```

Run   Task3Kt

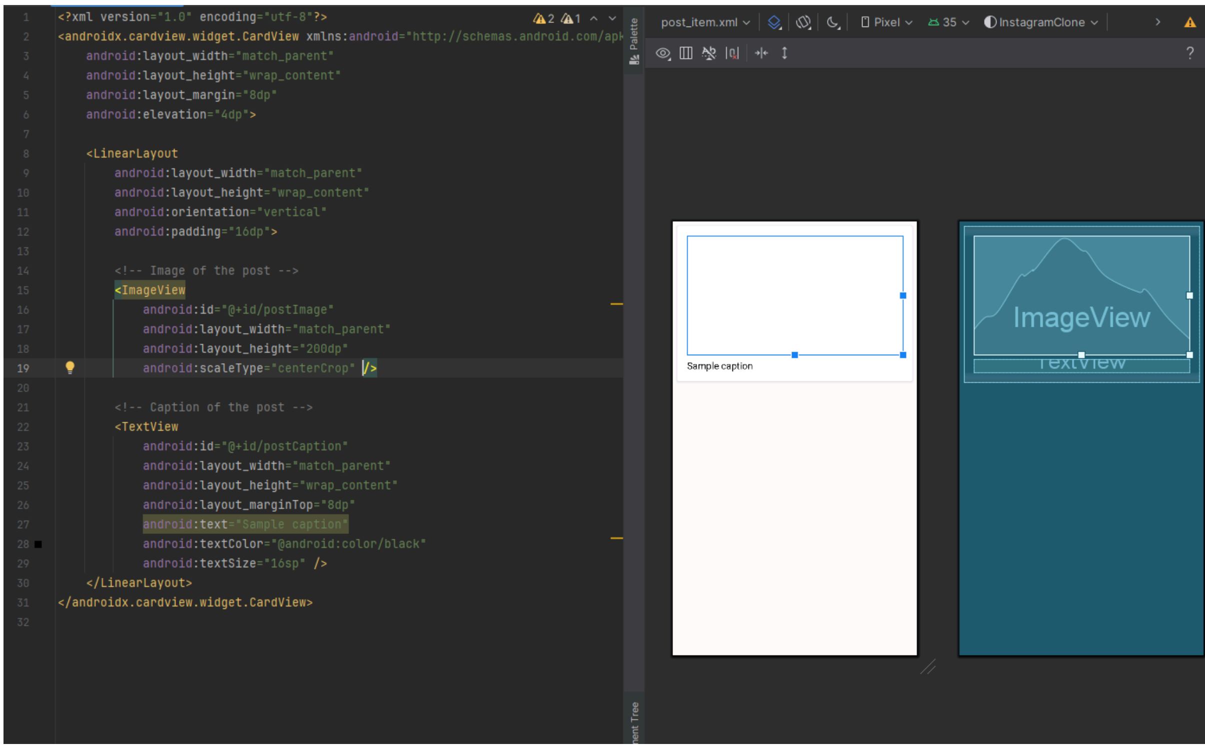
```
C:\Users\baimu\.jdks\corretto-11.0.24\bin\java.exe "-javaagent:C:\User
4000

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.



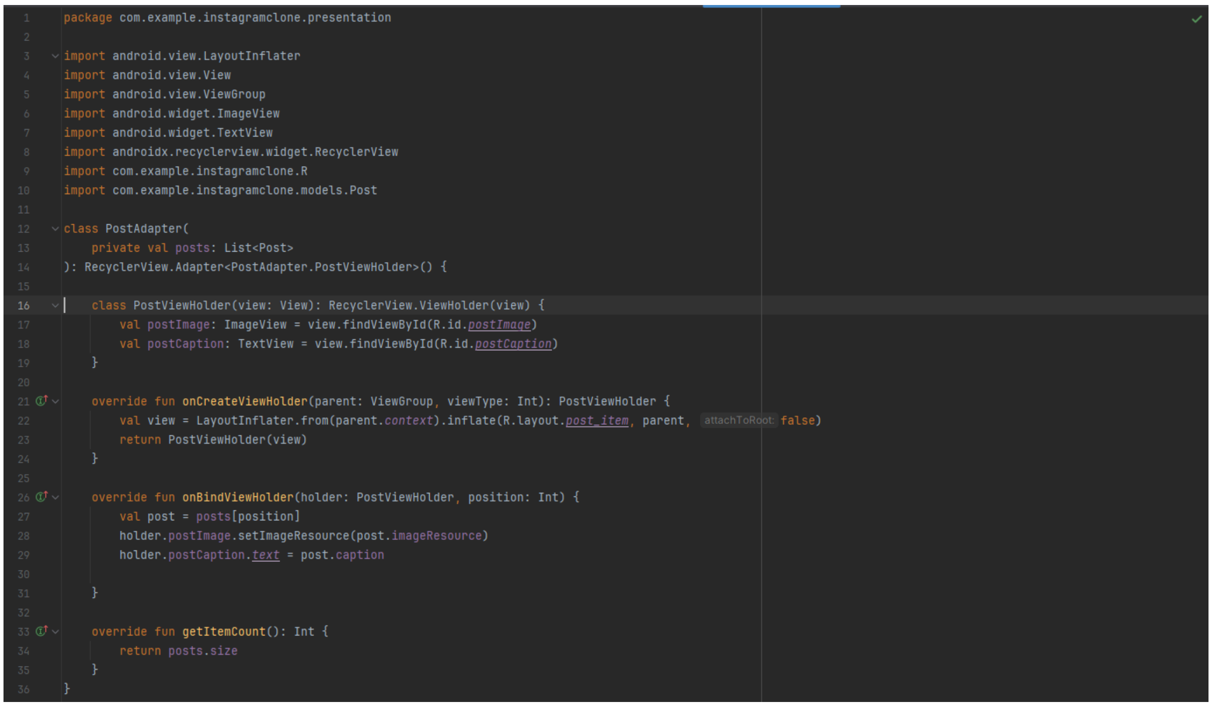
```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schema
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <!-- RecyclerView for the feed -->
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginTop="16dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

- ○ Include elements like `ImageView`, `TextView`, and `RecyclerView` for the feed
- ○ 

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp"
    android:elevation="4dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="16dp">

        <!-- Image of the post -->
        <ImageView
            android:id="@+id/postImage"
            android:layout_width="match_parent"
            android:layout_height="200dp"
            android:scaleType="centerCrop" />

        <!-- Caption of the post -->
        <TextView
            android:id="@+id/postCaption"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:text="Sample caption"
            android:textColor="@android:color/black"
            android:textSize="16sp" />
    </LinearLayout>
</androidx.cardview.widget.CardView>
```

Create the RecyclerView Adapter:

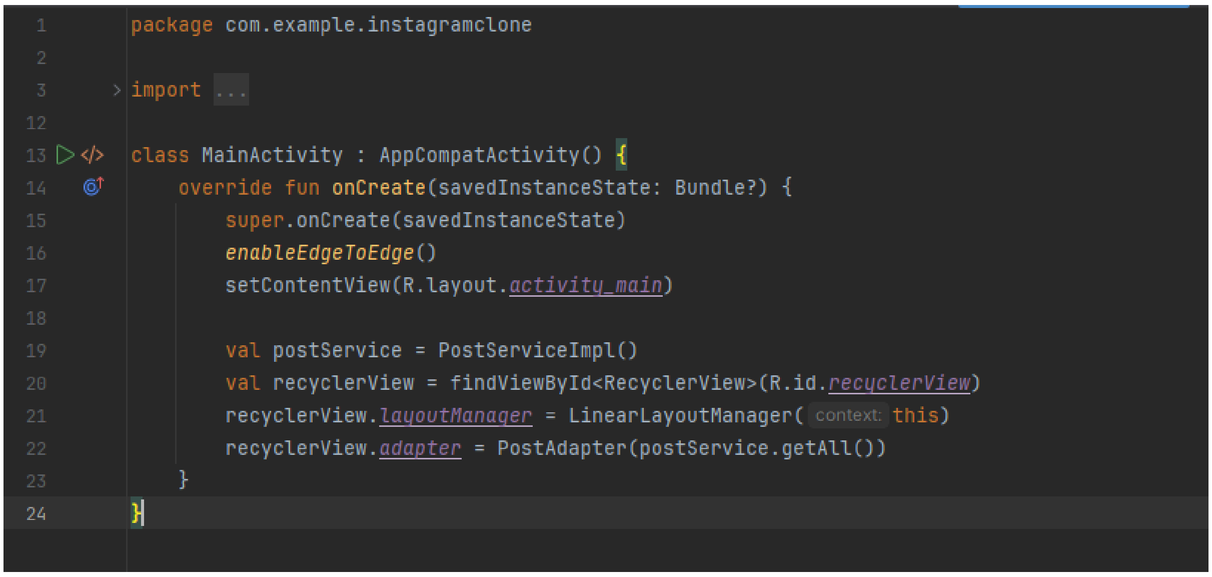
- Set up the RecyclerView to display a feed of posts with `ImageView` for the picture and `TextView` for the caption.

- 

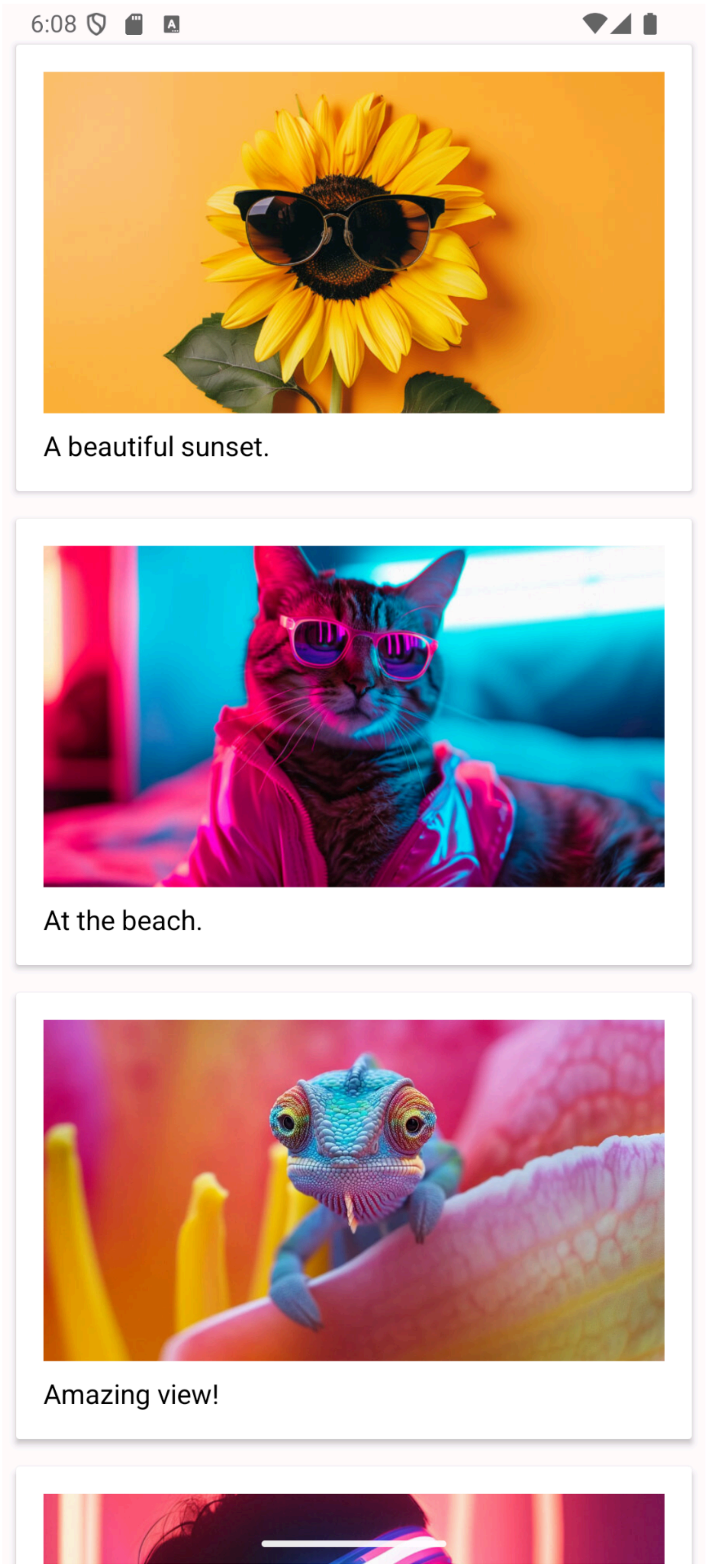
```kotlin
package com.example.instagramclone.presentation

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView
import com.example.instagramclone.R
import com.example.instagramclone.models.Post

class PostAdapter(
    private val posts: List<Post>
): RecyclerView.Adapter<PostAdapter.PostViewHolder>() {

    class PostViewHolder(view: View): RecyclerView.ViewHolder(view) {
        val postImage: ImageView = view.findViewById(R.id.postImage)
        val postCaption: TextView = view.findViewById(R.id.postCaption)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PostViewHolder {
        val view = LayoutInflater.from(parent.context).inflate(R.layout.post_item, parent,  attachToRoot: false)
        return PostViewHolder(view)
    }

    override fun onBindViewHolder(holder: PostViewHolder, position: Int) {
        val post = posts[position]
        holder.postImage.setImageResource(post.imageResource)
        holder.postCaption.text = post.caption

    }

    override fun getItemCount(): Int {
        return posts.size
    }
}
```

MainActivity Setup:

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



```kotlin
package com.example.instagramclone

import ...

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)

        val postService = PostServiceImpl()
        val recyclerView = findViewById<RecyclerView>(R.id.recyclerView)
        recyclerView.layoutManager = LinearLayoutManager( context: this)
        recyclerView.adapter = PostAdapter(postService.getAll())
    }
}
```



A beautiful sunset.

At the beach.

Amazing view!