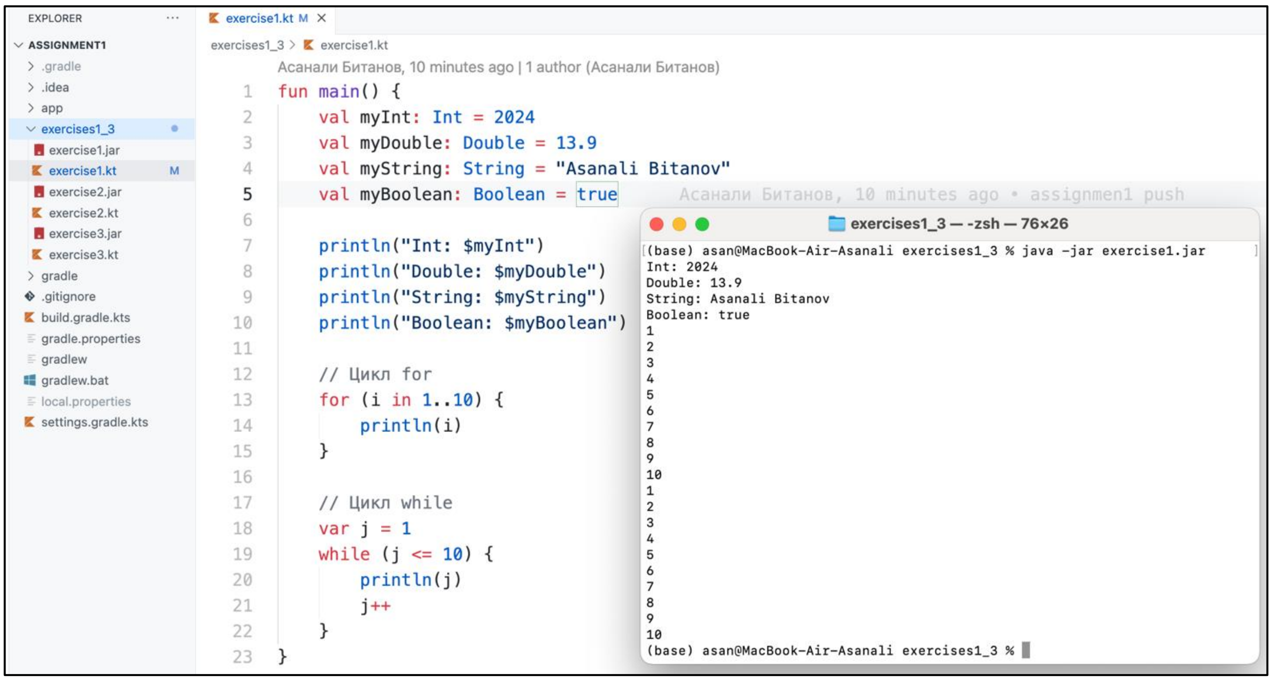
Mobile Development

Bitanov Asanali 23MD0392

Assignment 1

Exercise 1: Kotlin Syntax Basics

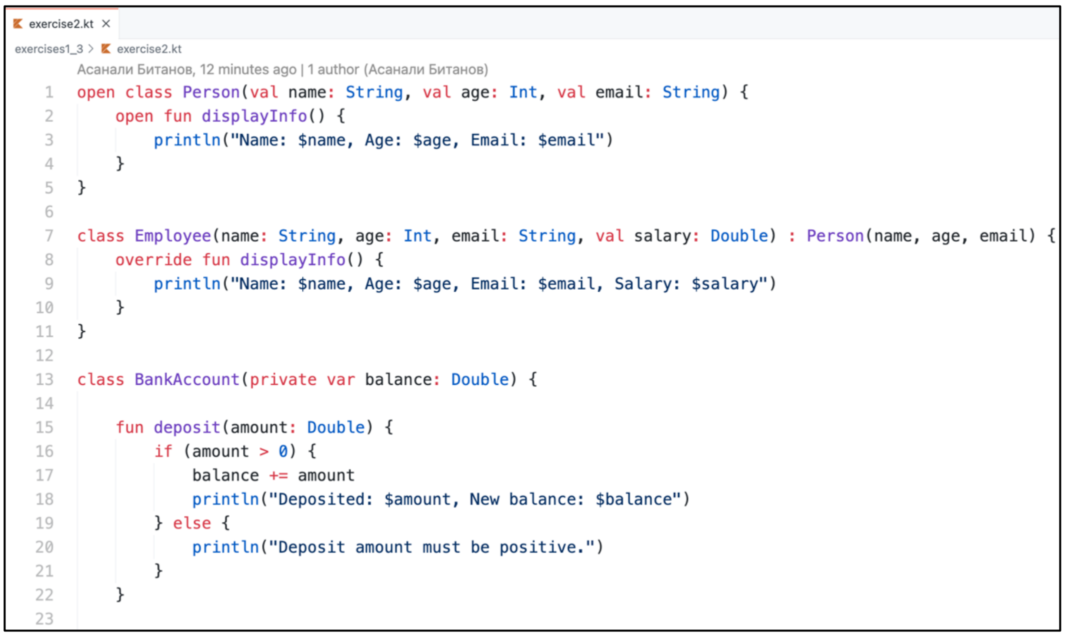


```kotlin
fun main() {
    val myInt: Int = 2024
    val myDouble: Double = 13.9
    val myString: String = "Asanali Bitanov"
    val myBoolean: Boolean = true

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

    // Цикл for
    for (i in 1..10) {
        println(i)
    }

    // Цикл while
    var j = 1
    while (j <= 10) {
        println(j)
        j++
    }
}
```

```
(base) asan@MacBook-Air-Asanali exercises1_3 % java -jar exercise1.jar
Int: 2024
Double: 13.9
String: Asanali Bitanov
Boolean: true
1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
10
(base) asan@MacBook-Air-Asanali exercises1_3 %
```

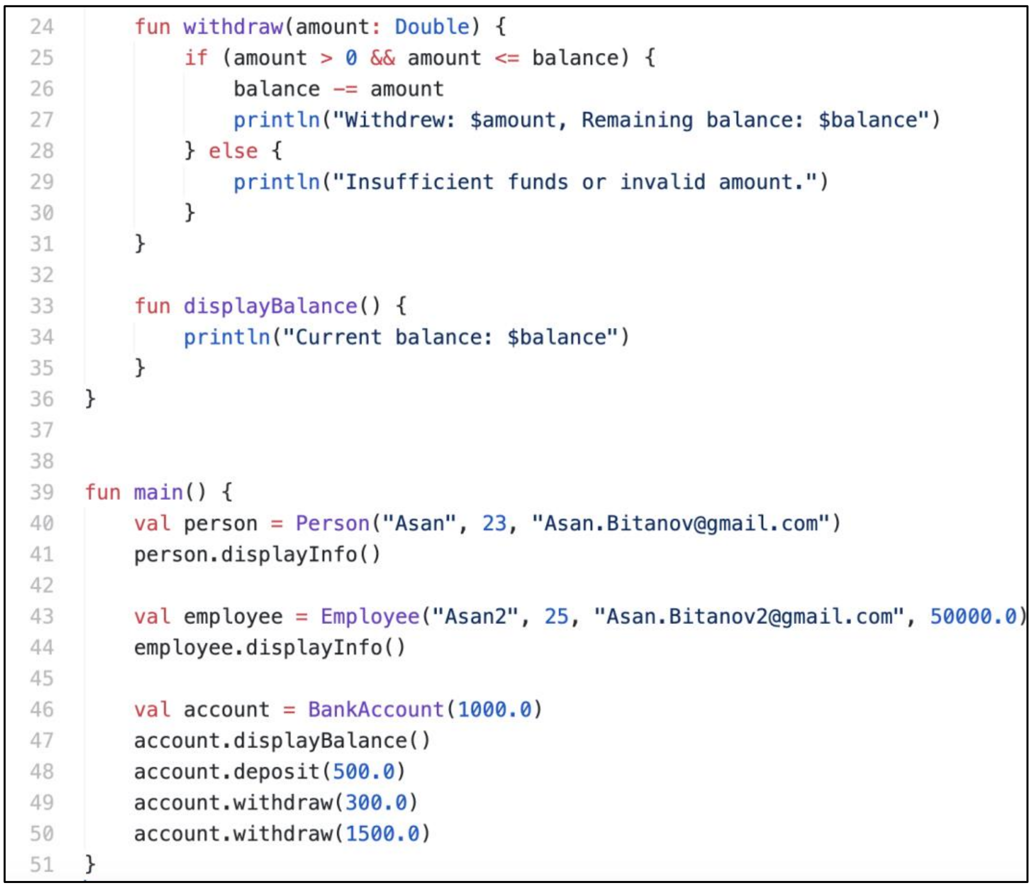
Fig 1. Exercise 1

Exercise 2: Kotlin OOP (Object-Oriented Programming)


exercise2.kt
exercises1_3 > exercise2.kt
Асанали Битанов, 12 minutes ago | 1 author (Асанали Битанов)

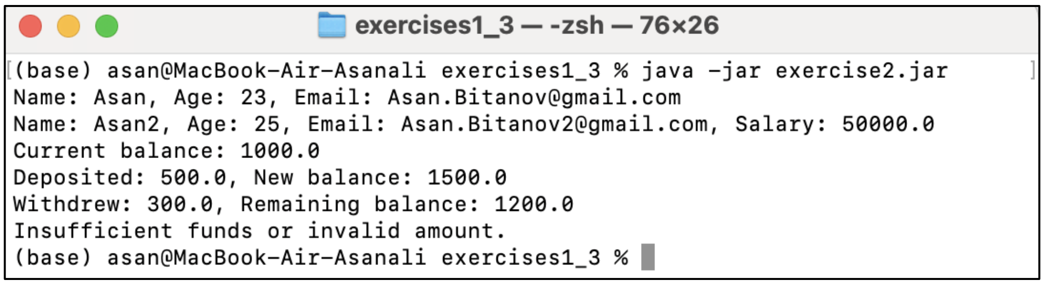
```kotlin
open class Person(val name: String, val age: Int, val email: String) {
    open fun displayInfo() {
        println("Name: $name, Age: $age, Email: $email")
    }
}

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

class BankAccount(private var balance: Double) {

    fun deposit(amount: Double) {
        if (amount > 0) {
            balance += amount
            println("Deposited: $amount, New balance: $balance")
        } else {
            println("Deposit amount must be positive.")
        }
    }

```

Fig 2. Exercise 2 code



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

    fun displayBalance() {
        println("Current balance: $balance")
    }
}


fun main() {
    val person = Person("Asan", 23, "Asan.Bitanov@gmail.com")
    person.displayInfo()

    val employee = Employee("Asan2", 25, "Asan.Bitanov2@gmail.com", 50000.0)
    employee.displayInfo()

    val account = BankAccount(1000.0)
    account.displayBalance()
    account.deposit(500.0)
    account.withdraw(300.0)
    account.withdraw(1500.0)
}
```

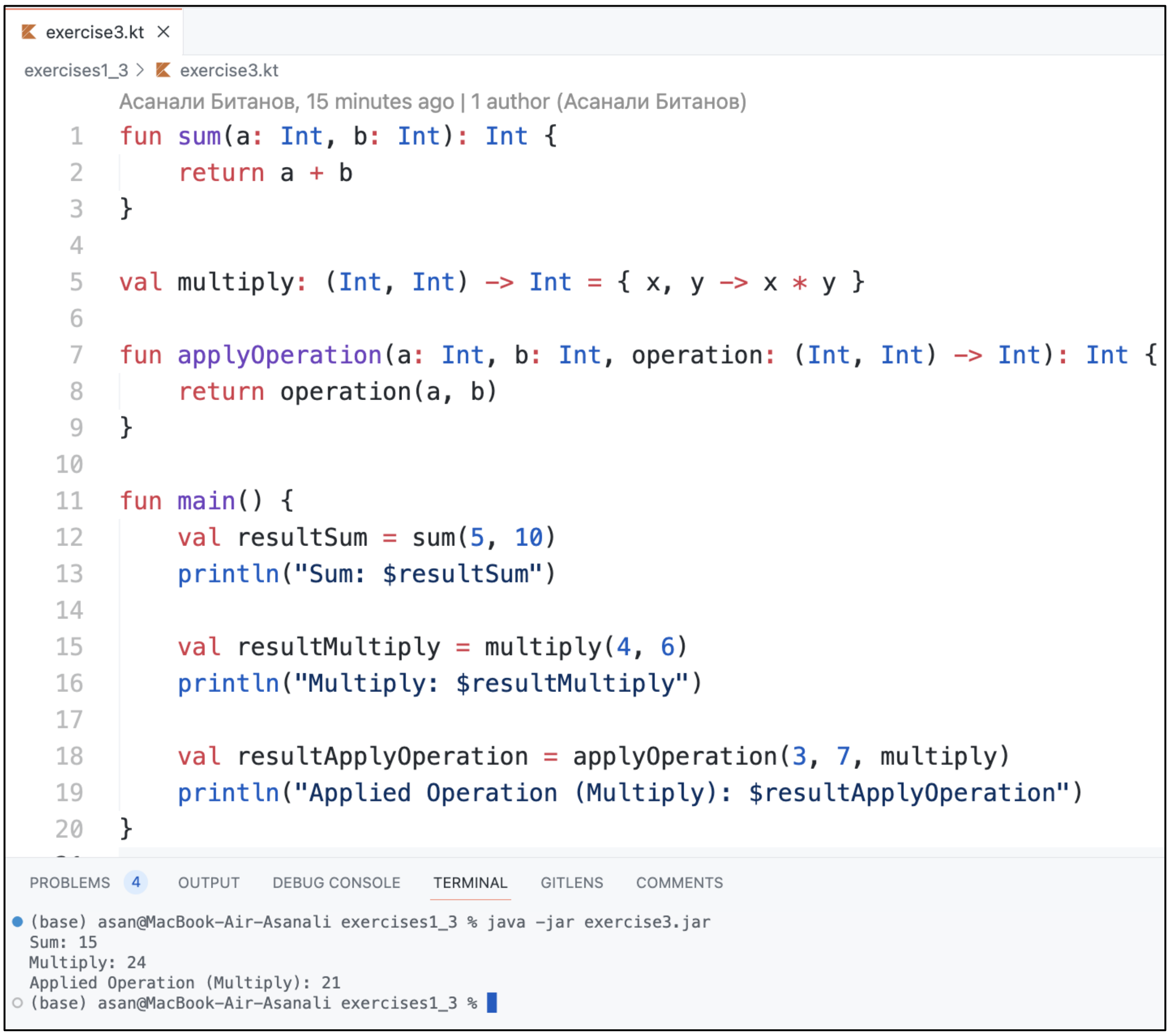
Fig 3. Exercise code part 2



```
(base) asan@MacBook-Air-Asanali exercises1_3 % java -jar exercise2.jar
Name: Asan, Age: 23, Email: Asan.Bitanov@gmail.com
Name: Asan2, Age: 25, Email: Asan.Bitanov2@gmail.com, Salary: 50000.0
Current balance: 1000.0
Deposited: 500.0, New balance: 1500.0
Withdrew: 300.0, Remaining balance: 1200.0
Insufficient funds or invalid amount.
(base) asan@MacBook-Air-Asanali exercises1_3 %
```

Fig 4. Exercise 2 output

Exercise 3: Kotlin Functions


exercise3.kt ×

exercises1_3 > exercise3.kt

Асанали Битанов, 15 minutes ago | 1 author (Асанали Битанов)

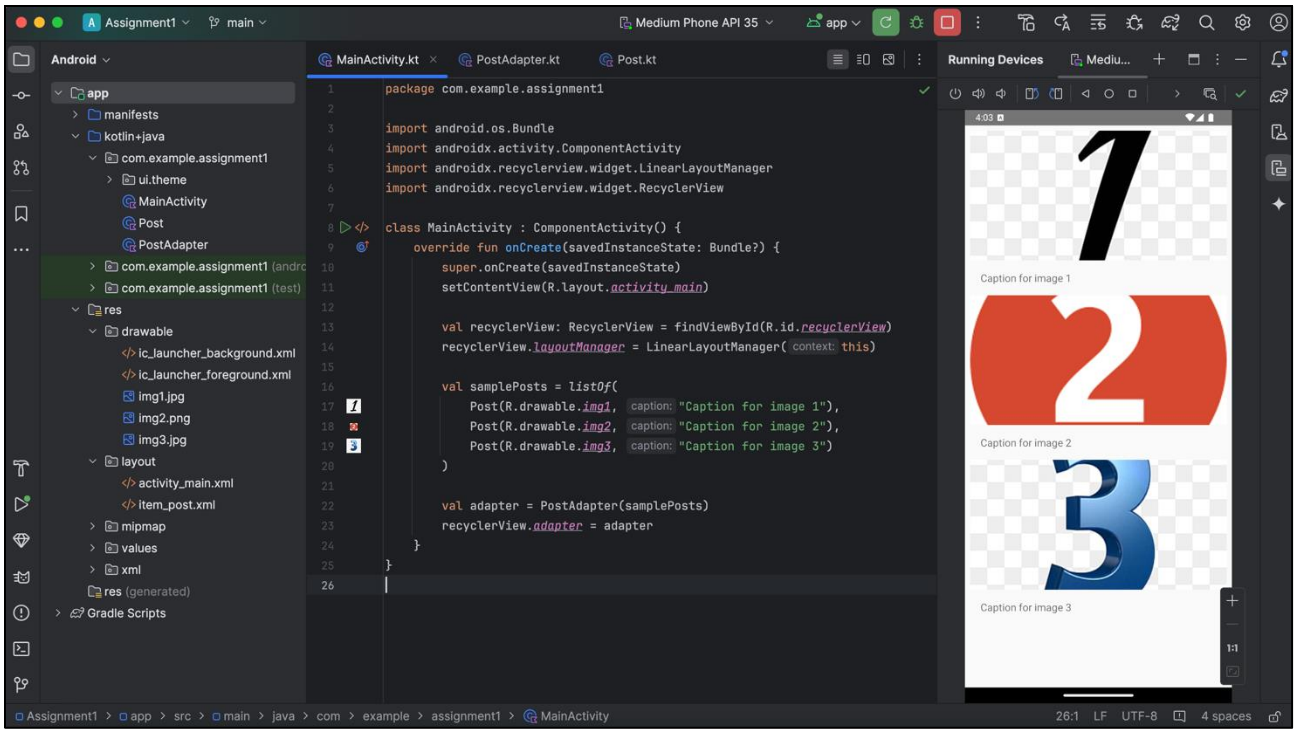
```kotlin
fun sum(a: Int, b: Int): Int {
    return a + b
}

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

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

fun main() {
    val resultSum = sum(5, 10)
    println("Sum: $resultSum")

    val resultMultiply = multiply(4, 6)
    println("Multiply: $resultMultiply")

    val resultApplyOperation = applyOperation(3, 7, multiply)
    println("Applied Operation (Multiply): $resultApplyOperation")
}
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL GITLENS COMMENTS

```
● (base) asan@MacBook-Air-Asanali exercises1_3 % java -jar exercise3.jar
  Sum: 15
  Multiply: 24
  Applied Operation (Multiply): 21
○ (base) asan@MacBook-Air-Asanali exercises1_3 %
```

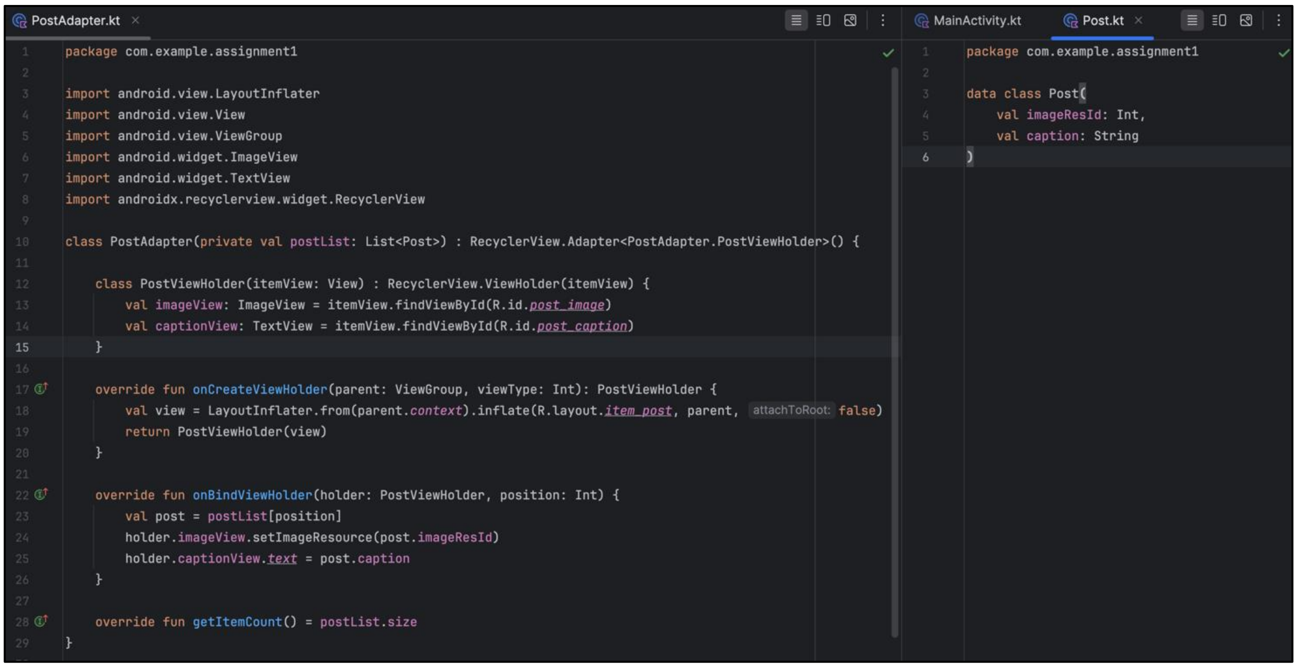
Fig 5. Exercise 3

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



```kotlin
package com.example.assignment1

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView

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

        val recyclerView: RecyclerView = findViewById(R.id.recyclerView)
        recyclerView.layoutManager = LinearLayoutManager( context: this)

        val samplePosts = listOf(
            Post(R.drawable.img1,  caption: "Caption for image 1"),
            Post(R.drawable.img2,  caption: "Caption for image 2"),
            Post(R.drawable.img3,  caption: "Caption for image 3")
        )

        val adapter = PostAdapter(samplePosts)
        recyclerView.adapter = adapter
    }
}
```

Fig 6. MainActivity.kt



```kotlin
package com.example.assignment1

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

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

    class PostViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        val imageView: ImageView = itemView.findViewById(R.id.post_image)
        val captionView: TextView = itemView.findViewById(R.id.post_caption)
    }

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

    override fun onBindViewHolder(holder: PostViewHolder, position: Int) {
        val post = postList[position]
        holder.imageView.setImageResource(post.imageResId)
        holder.captionView.text = post.caption
    }

    override fun getItemCount() = postList.size
}
```

```kotlin
package com.example.assignment1

data class Post(
    val imageResId: Int,
    val caption: String
)
```

Fig 7. PostAdapter.kt and Post.kt