Assignment 1 - Mobile Programming - Nurzhas Nurbayev

Exercise 1: Kotlin Syntax Basics

```kotlin
fun main() {

    val myInt: Int = 7
    val myDouble: Double = 7.99
    val myString: String = "Kotlin is cool!"
    val myBoolean: Boolean = true

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

```
Integer: 7
Double: 7.99
String: Kotlin is cool!
Boolean: true
```

Conditional Statements:

```kotlin
fun main() {
    val number = -5

    when {
        number > 0 -> println("$number is positive")
        number < 0 -> println("$number is negative")

        else -> println("$number is zero")
    }
}
```

Loops:

```kotlin
fun main() {

    println("Numbers using for loop:")

    for (i in 1..10) {
        print(i)
    }

    println()
    println("Numbers using while loop:")

    var i = 1

    while (i <= 10) {
        print(i)
        i++
    }
}
```

```
Numbers using for loop:
12345678910
Numbers using while loop:
12345678910
```

Collections:

```
fun main() {
    val numbers = listOf(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

    var sum = 0

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

    println("Sum is : $sum")
}
```

```
Sum is : 55
```

Exercise 2: Kotlin OOP (Object-Oriented Programming)

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

fun main() {
    val person = Person("Nurzhas", 22, "n_nurbayev@kbtu.kz")

    person.displayDetails()
}
```

Inheritance:

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

class Employee(name: String, age: Int, email: String, val salary: Double)
    : Person(name, age, email) {

    override fun displayDetails() {
        super.displayDetails()
        println("Salary: $salary")
    }
}
```

```
fun main() {
    val employee = Employee("Nurzhas", 22, "n_nurbayev@kbtu.kz", 900.000)

    employee.displayDetails()
}
```

```
Name: Nurzhas
Age: 22
Email: n_nurbayev@kbtu.kz
Salary: 900.0
```

Encapsulation:

```
class BankAccount(private var balance: Double) {
    fun deposit(amount: Double) {
        if (amount > 0) {
            balance += amount
            println("Successfully deposited: $amount")
        } else {
            println("Invalid deposit amount!")
        }
    }

    fun withdraw(amount: Double) {
        if (amount > 0 && amount <= balance) {
            balance -= amount
            println("Successfully withdrew: $amount")
        } else {
            println("Insufficient balance or invalid withdrawal amount!")
        }
    }

    fun checkBalance() {
        println("Current balance: $balance")
    }
}
```

```
fun main() {
    val account = BankAccount(50.0)

    account.checkBalance()
    account.deposit(10.0)
    account.checkBalance()
    account.withdraw(30.0)
    account.checkBalance()
    account.withdraw(150.0)
}
```

```
Current balance: 50.0
Successfully deposited: 10.0
Current balance: 60.0
Successfully withdrew: 30.0
Current balance: 30.0
Insufficient balance or invalid withdrawal amount!
```

Exercise 3: Kotlin Functions

Basic Function:

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

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

```
Sum: 15
```

Lambda Functions:

```kotlin
val multiply: (Int, Int) -> Int = { a, b -> a * b }

fun main() {
    val result = multiply(2, 3)
    println("Res: $result")
}
```

```
Res: 6
```

Higher-Order Functions:

```kotlin
fun applyFunc(a: Int, b: Int, func: (Int, Int) -> Int): Int {
    return func(a, b)
}

fun main() {
    val sumResult = applyFunc(5, 10) { x, y -> x + y }
    println("Sum res: $sumResult")

    val multiplyResult = applyFunc(5, 10) { x, y -> x * y }
    println("Mult res: $multiplyResult")
}
```

```
Sum res: 15
Mult res: 50
```

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