Assignment 1, Mobile Programming

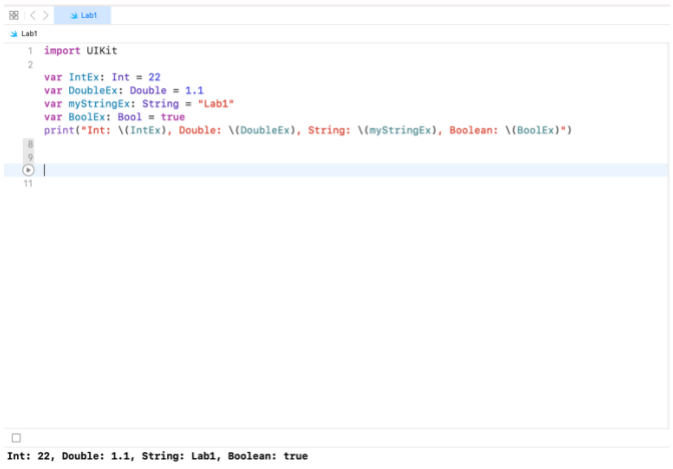
Aliyev Ramazan

Put all deliverables into github repository in your profile. Share link to google form 24 hours before defense. Defend by explaining deliverables and answering questions. There should be proof that you did yourself.
Deliverables: report in pdf
Google form: https://docs.google.com/forms/d/e/1FAIpQLSe0GyNdOYlvM1tX_I_CtlPod5jBf-ACLGdHYZq1gVZbUeBzIg/viewform?usp=sf_link

Git: https://github.com/rvmzik/Lab1_layout_task.git

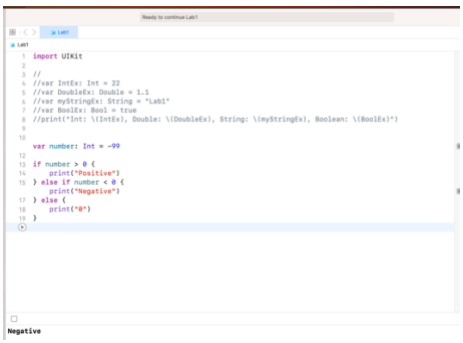
Exercise 1: Kotlin Syntax Basics

1. **Variables and Data Types:**
    - Create variables of different data types: `Int`, `Double`, `String`, `Boolean`.
    - Print the variables using `println`.
    - I did it with swift



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

```
let n = [1,2,3,4,5,6,7,8,9,10]

for i in 0..<n.count {
    print(n[i])
}
```

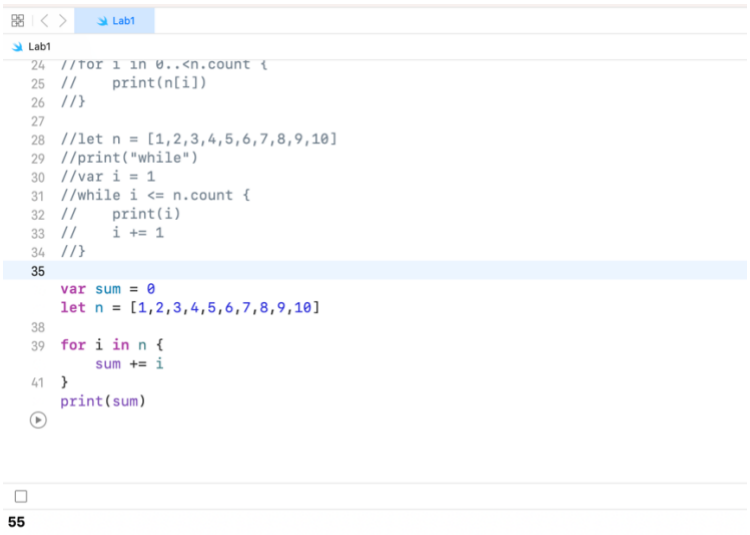
```
1
2
3
4
5
6
7
8
9
10
```

```
let n = [1,2,3,4,5,6,7,8,9,10]
print("while")
var i = 1
while i <= n.count {
    print(i)
    i += 1
}
```

```
while
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.



Exercise 2: Kotlin OOP (Object-Oriented Programming)

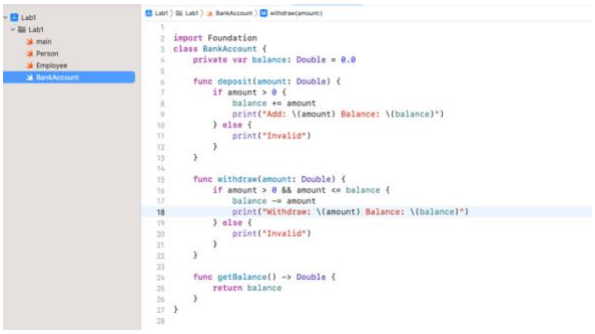
1. **Create a `Person` class:**
    - Define properties for `name`, `age`, and `email`.
    - Create a method to display the person's details.

Inheritance:

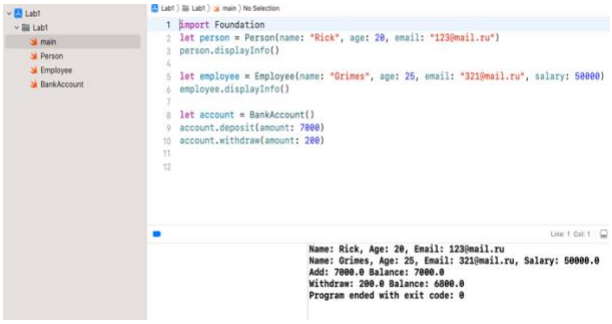
- Create a class `Employee` that inherits from the `Person` class.
- Add a property for `salary`.
- Override the `displayInfo` method to include the salary.

Encapsulation:

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



```
import Foundation
class BankAccount {
    private var balance: Double = 0.0

    func deposit(amount: Double) {
        if amount > 0 {
            balance += amount
            print("Add: \(amount) Balance: \(balance)")
        } else {
            print("Invalid")
        }
    }

    func withdraw(amount: Double) {
        if amount > 0 && amount <= balance {
            balance -= amount
            print("Withdraw: \(amount) Balance: \(balance)")
        } else {
            print("Invalid")
        }
    }

    func getBalance() -> Double {
        return balance
    }
}
```

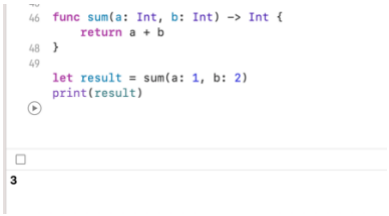


```
import Foundation
let person = Person(name: "Rick", age: 20, email: "123@mail.ru")
person.displayInfo()

let employee = Employee(name: "Grimes", age: 25, email: "321@mail.ru", salary: 50000)
employee.displayInfo()

let account = BankAccount()
account.deposit(amount: 7000)
account.withdraw(amount: 200)
```

```
Name: Rick, Age: 20, Email: 123@mail.ru
Name: Grimes, Age: 25, Email: 321@mail.ru, Salary: 50000.0
Add: 7000.0 Balance: 7000.0
Withdraw: 200.0 Balance: 6800.0
Program ended with exit code: 0
```

Exercise 3: Kotlin Functions

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

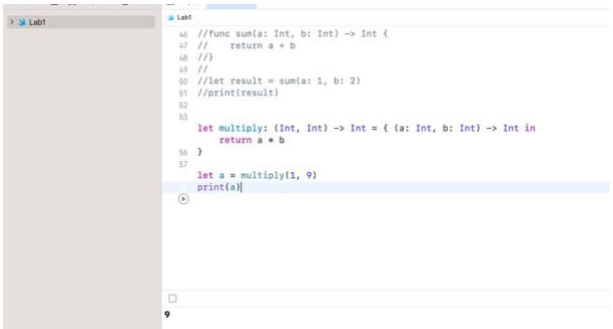


```
func sum(a: Int, b: Int) -> Int {
    return a + b
}

let result = sum(a: 1, b: 2)
print(result)
```

```
3
```

Lambda Functions:

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

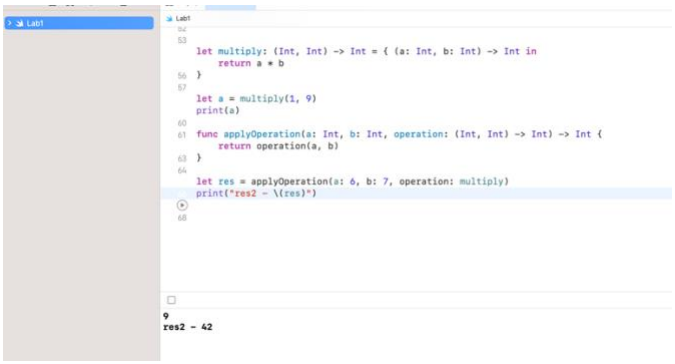


```
//func sum(a: Int, b: Int) -> Int {
//    return a + b
//}
//
//let result = sum(a: 1, b: 2)
//print(result)


let multiply: (Int, Int) -> Int = { (a: Int, b: Int) -> Int in
    return a * b
}

let a = multiply(1, 9)
print(a)
```

```
9
```

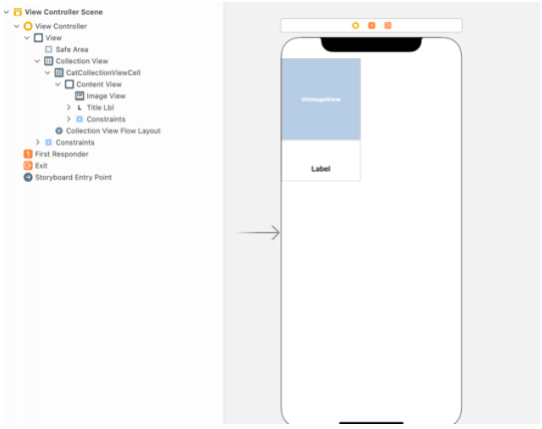
Higher-Order Functions:

- 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

  I use CollectionView in swift, it's like a RecyclerView