

Assignment 1, Mobile Programming

Bolatova Aruzhan(23MD0393)

Exercise 1: Kotlin Syntax Basics

Variables and Data Types:

```
aru_task1.kt x
1 //ex1
2
3 fun main() {
4     val age: Int = 22
5     val doubb: Double = 3.14
6     val name: String = "ARUZHAN!"
7     val bl: Boolean = true
8
9     println("Integer: $age")
10    println("Double: $doubb")
11    println("String: $name")
12    println("Boolean: $bl")
13
Aru_task1Kt x
```

```
Integer: 22
Double: 3.14
String: ARUZHAN!
Boolean: true
```

Conditional Statements:

```
xam 14 checkNum( number: 5)
hem 15 printNUMfor()
inAc 16 printNUMwh()
xam 17 collectnum()
ampl 18
xam 19
sk1.k 20 }
ratec 21 fun checkNum(number: Int) {
ts 22     when {
23         number > 0 -> println("$number is positive")
24         number < 0 -> println("$number is negative")
25         else -> println("$number is zero")
26     }
27 }
28
29 fun printNUMfor() {
30     for (i in 1 .. 10) {
Aru_task1Kt x
```

5 is positive

Loops:

```
aru_task1.kt x
28
29 fun printNUMfor() {
30     for (i in 1 .. 10) {
31         println(i)
32     }
33 }
34
35 fun printNUMwh() {
36     var i = 1
37     while (i <= 10) {
38         println(i)
39         i++
40     }
41 }
42 fun collectnum() {
43     val numbers = listOf(1, 2, 3, 4, 5)
44     var sum = 0
45     for (number in numbers) {
```



Collections:

```

}

fun collectnum() {
    val numbers = listOf(1, 2, 3, 4, 5)
    var sum = 0
    for (number in numbers) {
        sum += number
    }
    println("Sum: $sum")
}

```

Sum: 15

Exercise 2: Kotlin OOP (Object-Oriented Programming)

```

aru_task2.kt x
1 fun main() {
2     val person = Person(name: "Aruzhan Bolatova", age: 22, email: "aru.b@gmail.com")
3     person.displayInfo()
4     val employee = Employee(name: "Jay Park", age: 24, email: "jay.park@gmail.com", salary: 50000.0)
5     employee.displayInfo()
6     val account = BankAccount(balance: 1000.0)
7     account.deposit(amount: 500.0)
8     account.withdraw(amount: 300.0)
9     account.withdraw(amount: 1500.0) // Should not allow
10 }
11
12 open class Person(val name: String, val age: Int, val email: String) {
13     open fun displayInfo() {
14         println("Name: $name, Age: $age, Email: $email")
15     }
16 }
17 class Employee(name: String, age: Int, email: String, val salary: Double) : Person(name, age, email) {
18     override fun displayInfo() {
19         println("Name: $name, Age: $age, Email: $email, Salary: $salary")
20     }
21 }
22
23 class BankAccount(private var balance: Double) {
24

```

```

aru_task2.kt x
23 class BankAccount(private var balance: Double) {
24
25     fun deposit(amount: Double) {
26         if (amount > 0) {
27             balance += amount
28             println("Deposited $amount, New Balance: $balance")
29         }
30     }
31
32     fun withdraw(amount: Double) {
33         if (amount > 0 && balance >= amount) {
34             balance -= amount
35             println("Withdrew $amount, New Balance: $balance")
36         } else {
37             println("Insufficient funds or invalid amount.")
38         }
39     }
40
41     fun getBalance(): Double {
42         return balance
43     }
44 }
45

```

```

app x Aru_task2Kt x
| :
"C:\Program Files\Android\Android Studio\jbr\bin\java.exe" ...
Name: Aruzhan Bolatova, Age: 22, Email: aru.b@gmail.com
Name: Jay Park, Age: 24, Email: jay.park@gmail.com, Salary: 50000.0
Deposited 500.0, New Balance: 1500.0
Withdrew 300.0, New Balance: 1200.0
Insufficient funds or invalid amount.

Process finished with exit code 0

aruzhan > app > src > main > java > aru_task2.kt > BankAccount >

```

Exercise 3: Kotlin Functions

```
aru_task3.kt x
1  fun main() {
2      val sumResult = sum(a: 3, b: 5)
3      println("Сумма 3 и 5: $sumResult")
4
5
6      val multiply = { a: Int, b: Int -> a * b }
7      val multiplicationResult = multiply(4, 6)
8      println("Произведение 4 и 6: $multiplicationResult")
9
10
11     val resultHigherOrder = applyOperation(a: 10, b: 20, multiply)
12     println("Результат применения лямбда-функции: $resultHigherOrder")
13 }
14
15
16 fun sum(a: Int, b: Int): Int {
17     return a + b
18 }
19
20
21 fun applyOperation(a: Int, b: Int, operation: (Int, Int) -> Int): Int {
22     return operation(a, b)
23 }
```

```
an  Version control  Pixel Fold API 35  Aru_task3Kt  ▶  ⚙  □  ⋮
aru_task3.kt x
1  fun main() {
2      println("Произведение 4 и 6: $multiplicationResult")
3
4
5
6
7
8
9
10
11     val resultHigherOrder = applyOperation(a: 10, b: 20, multiply)
12     println("Результат применения лямбда-функции: $resultHigherOrder")
13 }
14
15
16 fun sum(a: Int, b: Int): Int {
17     return a + b
18 }
19
20
21 fun applyOperation(a: Int, b: Int, operation: (Int, Int) -> Int): Int {
22     return operation(a, b)
23 }
24
```

app × Aru_task3Kt ×

"C:\Program Files\Android\Android Studio\jbr\bin\java.exe" ...

Сумма 3 и 5: 8

Произведение 4 и 6: 24

Результат применения лямбда-функции: 200

Process finished with exit code 0

|