Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Дисциплина: Программирование мобильных информационных систем

Отчёт

по лабораторной работе №1

на тему:

**«Основы языка Kotlin»**

Выполнил:

ст. гр. 214302

Кранцевич В.С.

Проверил:

Усенко Ф. В.

Минск 2024

**Цель**: Изучить основные конструкции языка Kotlin, научиться работать с типами данных, операциями ввода-вывода, а также условными операторами. Закрепить полученные знания через выполнение задач, требующих практического применения теории.

**Вариант 17**: Калькулятор с решением уравнений: Создайте программу, которая решает линейные уравнения вида ax + b = 0 и выводит корень уравнения. Программа должна проверять ввод пользователя и обрабатывать исключительные случаи (например, когда коэффициент a равен нулю).

**КОД ПРОГРАММЫ**

import java.util.Scanner

fun main() {

val scanner = Scanner(System.`in`)

try {

for (a in 1..10) {

if (a % 2 == 0){

println("Число чётное - $a")

}

else {

println ("Число нечётное - $a")

}

}

print("Введите коэффициент a: ")

val a = scanner.nextDouble()

print("Введите коэффициент b: ")

val b = scanner.nextDouble()

if (a == 0.0) {

if (b == 0.0) {

println("Уравнение имеет бесконечное множество решений.")

} else {

println("Уравнение не имеет решений, так как a = 0 и b != 0.")

}

} else {

val x = -b / a

println("Корень уравнения: x = $x")

}

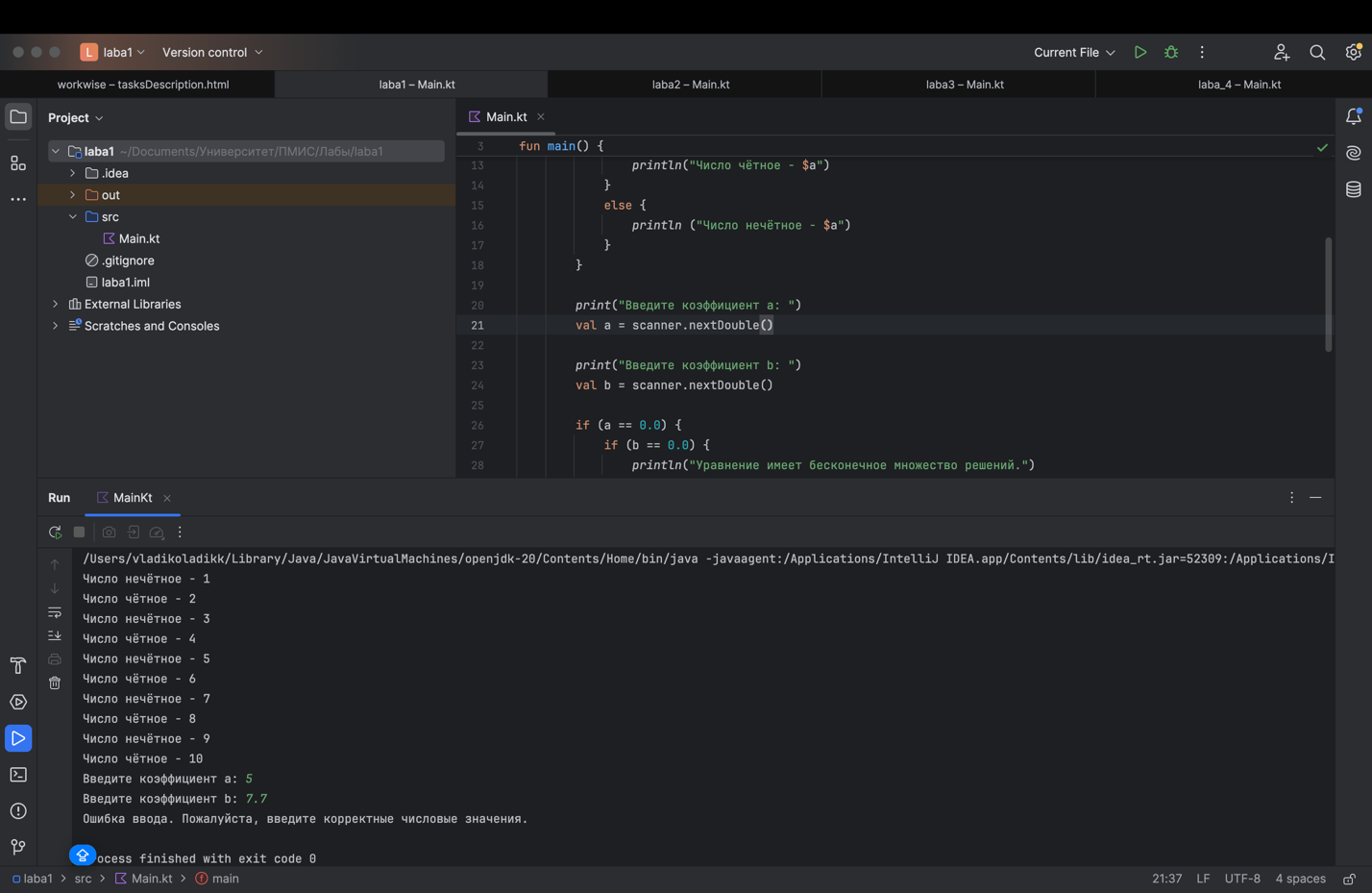
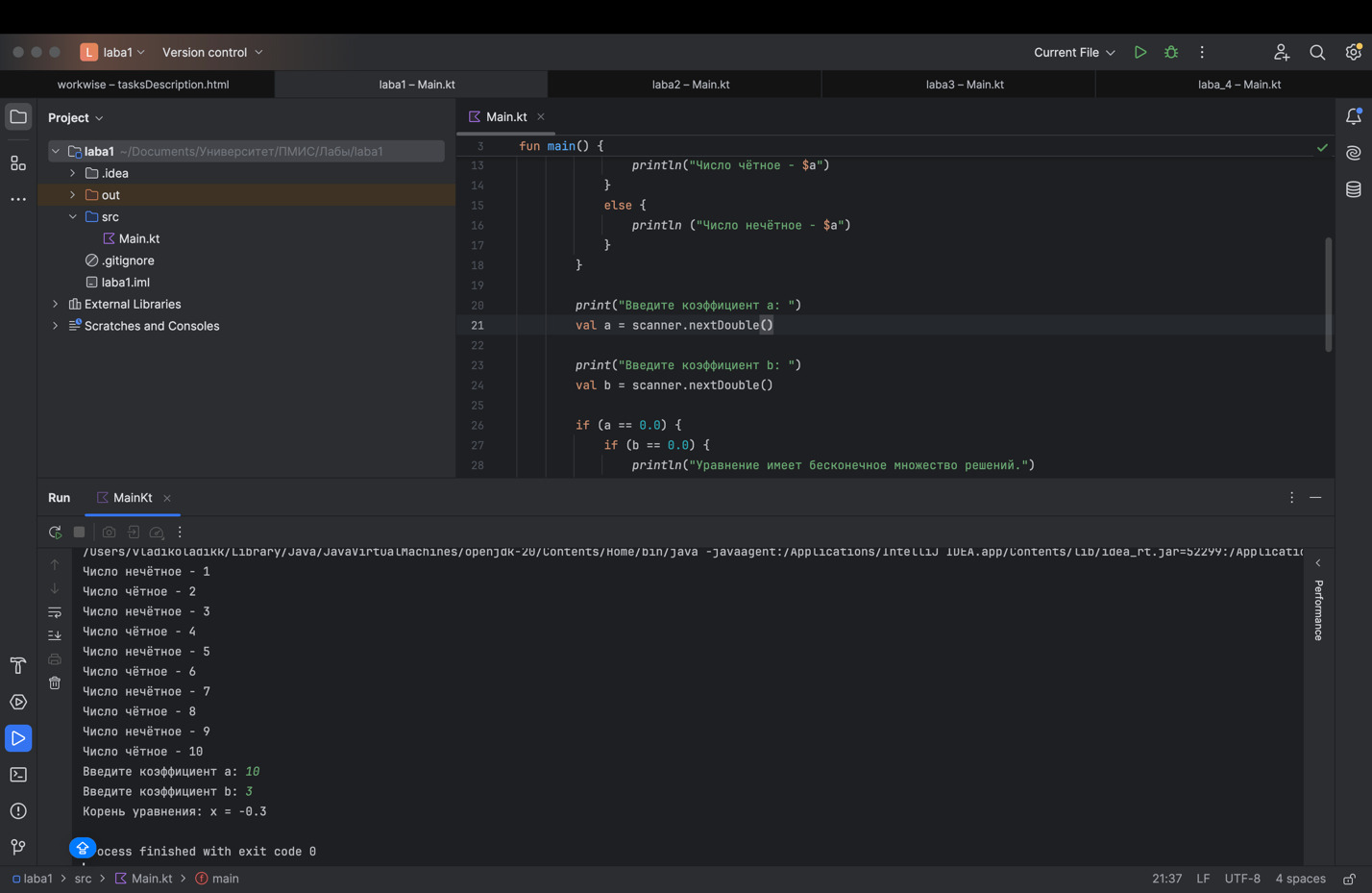
} catch (e: Exception) {

println("Ошибка ввода. Пожалуйста, введите корректные числовые значения.")

}

}

**СКРИНШОТЫ РАБОЧИХ ОКОН**

****

**ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ**

**1. Что такое функция main в Kotlin, и какова её роль в программе?**

Функция main в Kotlin — это точка входа в программу. Именно с неё начинается выполнение приложения. Она имеет следующий синтаксис:

fun main() {

println("Hello, Kotlin!")

}

Роль функции main — запуск программы и управление её выполнением.

**2. Чем отличаются переменные, объявленные с использованием val и var?**

* **val**: Используется для объявления неизменяемых переменных (аналог final в Java). Значение переменной нельзя изменить после присвоения.
* **var**: Используется для объявления изменяемых переменных. Значение переменной можно менять.

Пример:

val name = "Kotlin" // Нельзя изменить

var age = 25 // Можно изменить

age = 26

**3. Какие типы данных существуют в Kotlin для целых чисел? Приведите примеры.**

Kotlin предоставляет следующие типы данных для целых чисел:

* Byte (8 бит): от -128 до 127
* Short (16 бит): от -32,768 до 32,767
* Int (32 бита): от -2,147,483,648 до 2,147,483,647
* Long (64 бита): от -2^63 до 2^63-1

Пример:

val byteValue: Byte = 127

val shortValue: Short = 32000

val intValue: Int = 2147483647

val longValue: Long = 9223372036854775807L

**4. Как объявить строковую переменную в Kotlin? Можно ли её изменить после объявления?**

Объявить строковую переменную можно так:

val greeting: String = "Hello"

Если используется val, строку изменить нельзя. Если var, то можно:

var name = "Kotlin"

name = "Java" // Изменение строки

**5. Что такое интерполяция строк, и как она работает в Kotlin?**

Интерполяция строк позволяет вставлять значения переменных внутрь строки с использованием символа $.

Пример:

val name = "Kotlin"

val greeting = "Hello, $name!"

println(greeting) // Вывод: Hello, Kotlin!

Для выражений используется синтаксис ${}:

val age = 25

println("Возраст через 5 лет: ${age + 5}")

**6. Как в Kotlin можно обработать возможность появления null?**

Kotlin предотвращает ошибки, связанные с null, с помощью оператора ?. Переменные могут быть объявлены как nullable (Тип?) или non-nullable (Тип).

Пример:

val nullableString: String? = null

val length = nullableString?.length // Возвращает null, если строка равна null

Для задания значения по умолчанию используется оператор ?::

val length = nullableString?.length ?: 0 // Возвращает 0, если строка равна null

**7. Как организован ввод данных с консоли в Kotlin?**

Для ввода данных с консоли используется функция readLine().

Пример:

println("Введите имя:")

val name = readLine()

println("Привет, $name!")

**8. Чем отличаются функции print() и println()?**

* **print()**: Выводит текст без перевода строки.
* **println()**: Выводит текст и добавляет перевод строки.

Пример:

print("Hello")

print("World") // Вывод: HelloWorld

println("Hello")

println("World") // Вывод: Hello (с новой строки) World

**9. Как работают арифметические операторы в Kotlin? Приведите примеры.**

Арифметические операторы в Kotlin:

* + (сложение)
* - (вычитание)
* \* (умножение)
* / (деление)
* % (остаток от деления)

Пример:

val a = 10

val b = 3

println(a + b) // 13

println(a - b) // 7

println(a \* b) // 30

println(a / b) // 3

println(a % b) // 1

**10. Какие операторы используются для сравнения чисел в Kotlin?**

Операторы сравнения:

* == (равно)
* != (не равно)
* < (меньше)
* > (больше)
* <= (меньше или равно)
* >= (больше или равно)

Пример:

val a = 10

val b = 20

println(a < b) // true

println(a == b) // false

**11. Как работает конструкция if...else в Kotlin?**

Конструкция if...else используется для ветвления кода.

Пример:

val age = 18

if (age >= 18) {

println("Взрослый")

} else {

println("Ребенок")

}

**12. В чем разница между конструкциями if...else и when?**

* **if...else**: Используется для проверки условий.
* **when**: Используется для выбора из множества значений.

Пример when:

val day = 2

val result = when(day) {

1 -> "Понедельник"

2 -> "Вторник"

else -> "День неизвестен"

}

**13. Как создать функцию в Kotlin, которая возвращает значение? Приведите пример.**

Пример функции:

fun add(a: Int, b: Int): Int {

return a + b

}

println(add(5, 3)) // 8

**14. Что такое параметры функции по умолчанию, и как они используются в Kotlin?**

Функции могут иметь параметры с дефолтными значениями:

fun greet(name: String = "Гость") {

println("Привет, $name!")

}

greet() // Привет, Гость!

greet("Котлин") // Привет, Котлин!

**15. Можно ли передать функцию в качестве аргумента другой функции? Приведите пример.**

Да, функции — это объекты первого класса:

fun applyOperation(x: Int, y: Int, operation: (Int, Int) -> Int): Int {

return operation(x, y)

}

println(applyOperation(4, 5, ::add)) // 9

**16. Как Kotlin обрабатывает ошибки ввода данных?**

Для обработки ошибок используется try...catch:

try {

val number = readLine()!!.toInt()

println(number)

} catch (e: NumberFormatException) {

println("Ошибка ввода!")

}

**17. В чем разница между неизменяемыми и изменяемыми переменными?**

* **Изменяемые (var)**: Можно менять значение.
* **Неизменяемые (val)**: Нельзя менять значение после инициализации.

**18. Как организовать цикл для вывода всех чисел от 1 до 10? Приведите пример.**

for (i in 1..10) {

println(i)

}

**19. Какую конструкцию следует использовать для выбора действия на основе множества значений переменной?**

Используйте when.

**20. Как организовать проверку, является ли введенное число четным или нечетным? Приведите пример.**

val number = readLine()!!.toInt()

println(if (number % 2 == 0) "Четное" else "Нечетное")

**ВЫВОД**

В ходе лабораторной работы изучены основные конструкции языка *Kotlin*, типы данных, операций ввода-вывода, а также условные операторы. Закреплены полученные знания через выполнение задач, требующих практического применения теории.