Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

Отчёт

К лабораторной работе № 1

на тему:

**ОСНОВЫ ЯЗЫКА KOTLIN**

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Усенко Ф. В.

(подпись)

Выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.А. Грицков

(подпись) 214302

Минск 2024

**Вариант 6. Продвинутый калькулятор ИМТ с историей и графиками.**

Условие: Создайте программу, которая вычисляет ИМТ, сохраняет историю всех вычислений и строит график изменения ИМТ с течением времени. Программа должна поддерживать экспорт данных в текстовый файл.

**Код программы:**

import java.io.File  
import org.knowm.xchart.SwingWrapper  
import org.knowm.xchart.XYChartBuilder  
import org.knowm.xchart.style.markers.SeriesMarkers  
import java.text.NumberFormat  
import java.text.ParseException  
import java.util.\*  
  
fun main() {  
 var isAlive = true  
  
 println("Данная программа предназначена для расчета ИМТ ")  
  
 while (isAlive) {  
 println("1. Рассчитать ИМТ \n" +  
 "2. Вывести график \n" +  
 "3. Вывести данные из файла \n")  
 print("Выберите действие: ")  
  
 val input = readLine()  
  
 if (input != null) {  
 try {  
 val choice = input.toInt()  
  
 when (choice) {  
 1 -> inputData()  
 2 -> drawGraph()  
 3 -> readFromFile()  
 else -> {  
 println("Неверный выбор. Программа завершает работу...")  
 isAlive = false  
 }  
 }  
 } catch (e: NumberFormatException) {  
 println("Ошибка: Ввод должен быть числом. Пожалуйста, попробуйте снова.")  
 }  
 } else {  
 println("Ошибка: Ввод не может быть пустым. Пожалуйста, попробуйте снова.")  
 }  
 }  
}  
  
  
fun calculate\_BMI(mass: Double, height: Double): Double{  
 return mass/(height \* height)  
}  
fun analize\_BMI(bmi: Double){  
 when {  
 bmi < 18.5 -> println("Недостаточный вес")  
 bmi in 18.5..24.9 -> println("Нормальный вес")  
 bmi in 25.0..29.9 -> println("Избыточный вес")  
 else -> println("Ожирение")  
 }  
}  
  
fun saveInFile(weight: Double, height: Double, bmi: Double) {  
 val fileName = "BMI\_DATA.txt"  
 val data = "Вес: $weight кг. Рост: $height м. Индекс Массы Тела: %.2f\n".format(bmi)  
  
 try {  
 File(fileName).appendText(data + "\n\n")  
 println("Данные успешно сохранены в файл $fileName\n\n")  
 } catch (e: Exception) {  
 println("Ошибка при сохранении данных: ${e.message}\n\n")  
 }  
}  
  
fun readFromFile(){  
 try {  
 val lines = File("BMI\_DATA.txt").readLines()  
 if (lines.isNotEmpty()) {  
 println("\n\n=== Содержимое файла BMI\_DATA.txt ===\n\n")  
 lines.forEach { println(it) } // что такое it  
 println()  
 } else {  
 println("Файл \"BMI\_DATA.txt\" пуст.")  
 }  
 } catch (e: Exception) {  
 println("Ошибка при чтении файла: ${e.message}")  
 }  
}  
  
fun inputData(){  
 var weight: Double  
 var height: Double  
 var result: Double  
  
 while (true) {  
 print("Введите вес (в кг): ")  
 val weightInput = readLine()  
 if (weightInput != null && weightInput.toDoubleOrNull() != null && weightInput.toDouble() > 0) {  
 weight = weightInput.toDouble()  
 break  
 } else {  
 println("Неверный ввод. Вес должен быть положительным!")  
 }  
 }  
  
 while (true) {  
 print("Введите рост (в см): ")  
 val heightInput = readLine()  
 if (heightInput != null && heightInput.toDoubleOrNull() != null && heightInput.toDouble() > 0) {  
 height = heightInput.toDouble() / 100  
 break  
 } else {  
 println("Неверный ввод. Рост должен быть положительным!")  
 }  
 }  
 result = calculate\_BMI(weight, height)  
 println("Ваш Индекс Массы Тела: %.2f".format(result))  
 analize\_BMI(result)  
 saveInFile(weight, height, result)  
}  
fun drawGraph() {  
 val fileName = "BMI\_DATA.txt"  
 val bmiValues = mutableListOf<Double>()  
  
 try {  
 val lines = File(fileName).readLines()  
 lines.forEach { line ->  
 if (line.contains("Индекс Массы Тела:")) {  
 val bmiString = line.substringAfter("Индекс Массы Тела: ").trim()  
 try {  
 val numberFormat = NumberFormat.getInstance(Locale("ru", "RU"))  
 val bmi = numberFormat.parse(bmiString).toDouble()  
 bmiValues.add(bmi)  
 } catch (e: ParseException) {  
 println("Ошибка при парсинге значения ИМТ: $bmiString")  
 }  
 }  
 }  
  
 if (bmiValues.isEmpty()) {  
 println("Нет данных для построения графика.")  
 return  
 }  
  
 val chart = XYChartBuilder().width(800).height(600).title("График ИМТ").xAxisTitle("Измерение").yAxisTitle("ИМТ").build()  
  
 val xData = (1..bmiValues.size).map { it.toDouble() }  
 val yData = bmiValues  
 val series = chart.addSeries("ИМТ", xData, yData)  
 series.marker = SeriesMarkers.CIRCLE  
  
 chart.styler.yAxisMin = bmiValues.minOrNull()  
 chart.styler.yAxisMax = bmiValues.maxOrNull()  
  
 SwingWrapper(chart).displayChart()  
  
 } catch (e: Exception) {  
 println("Ошибка при построении графика: ${e.message}")  
 }  
}

**Контрольные вопросы:**

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

– Функция, являющаяся точкой входа в программу. В данной функции

выполняется основной код программы.

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

*var*?

– Обе конструкции используются для объявления переменных. Разница лишь в том, что *val* не дает переопределять значение переменной, в отличие от *var*.

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

– Для хранения целых данных есть *Int, Short, Byte, Long*. Для хранения дробных чисел есть *Float* и *Double*. Хранение символьных данных – *Char*. Логические значения – *Boolean*.

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

– Строковая переменная объявляется следующим образом: val greeting = "Hello". Для хранения строковых значений используется тип *String*. Данные такого типа изменять нельзя.

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

– Интерполяция – это альтернатива конкатенации в *Java*. Вместо присоединения переменных к строке (конкатенация), используется встраивание переменных в строки, что делает код более читаемым.

Пример: val greeting = "Привет, меня зовут $name и мне $age лет."