Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

Отчёт

по лабораторной работе №1

на тему:

**ОСНОВЫ ЯЗЫКА KOTLIN**

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

(подпись)

Выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. Н. Кубицкая

(подпись) гр. 214302

Минск, 2024

**Задание:** Напишите программу, которая переводит температуру из одной системы в другую и на основе исторических данных предсказывает температуру на следующий день.

**Решение:**

import java.io.File  
  
fun main() {  
 val fileName = "temperature\_data.txt"  
 val file = File(fileName) *//объект* while (true) {  
 *println*("Введите температуру (или '-' для завершения):")  
 val input = *readLine*()  
  
 if (input == "-") {  
 break  
 }  
  
 *// Проверка на корректность ввода числа* val temperature = input?.*toDoubleOrNull*() *//?.-проверяет не null ли... toDoubleOrNull() — это метод, который пытается преобразовать строку в число типа Double. Если преобразование не удается (например, если введена не числовая строка), метод возвращает null.* if (temperature == null) {  
 *println*("Некорректный ввод. Пожалуйста, введите число.")  
 continue  
 }  
  
 *println*("Введите систему (C для Цельсия, F для Фаренгейта):")  
 val system = *readLine*()!!.*toUpperCase*() *//Оператор !! используется для явного указания, что значение не будет null* if (system != "C" && system != "F") {  
 *println*("Некорректный ввод. Пожалуйста, введите 'C' или 'F'.")  
 continue  
 }  
  
 val convertedTemperature: Double  
  
 if (system == "C") {  
 convertedTemperature = *celsiusToFahrenheit*(temperature)  
 } else {  
 convertedTemperature = *fahrenheitToCelsius*(temperature)  
 }  
  
  
 *println*("Конвертированная температура: $convertedTemperature")  
  
 *// Запись данных в файл (только в С)* if (system=="C"){  
 file.*appendText*("$temperature\n")  
 } else file.*appendText*("$convertedTemperature\n")  
  
 *// Чтение данных из файла* val predictedTemperature : Double  
 val historicalData = file.*readLines*().*map* **{ it**.*toDouble*() **}** *// map преобразует каждый элемент коллекции* if (historicalData.size==1){  
 *println*("Нет исторических данных. Предсказание температуры на следующий день невозможно")  
 } else{  
 predictedTemperature = *predictNextDayTemperature*(historicalData)  
 *println*("Предсказанная температура на следующий день: $predictedTemperature")}  
 }  
}  
  
fun celsiusToFahrenheit(celsius: Double): Double {  
 return celsius \* 9/5 + 32  
}  
  
fun fahrenheitToCelsius(fahrenheit: Double): Double {  
 return (fahrenheit - 32) \* 5/9  
}  
  
fun predictNextDayTemperature(historicalData: List<Double>): Double {  
 *// Вычисляем все разности между соседними элементами* val differences = historicalData.*zipWithNext* **{** a, b **->** b - a **}** *//zipWithNext — это функция, которая создает пары соседних элементов списка  
  
 // Находим среднее значение разностей* val averageDifference = differences.*average*() *// average-среднее* return historicalData.*last*() + averageDifference *// Предсказываем следующее число*}

**Контрольные вопросы:**

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

Данная функция является точкой входа в программу. В этой функции выполняется основной код программы.

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

Переменные var являются изменяемыми, а val – неизменяемыми.

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

Для объявления строковой переменной используется тип String. Строку нельзя изменить, но можно создать новую строку на ее основе.

1. Чем отличаются функции print() и println()?

Функция println() добавляет перевод строки после вывода текста, тогда как print() этого не делает.