Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

Информатики и радиоэлектроники»

Учебная дисциплина: «Программирование мобильных информационных систем»

Отчет

по выполнению лабораторной работы «Основы языка Kotlin»

Выполнил студент: гр. 210101 Соболь М.А.

Проверил: Усенко Ф.В.

Минск 2024

**1. Цель работы:** 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.

**2. Задание:**Напишите программу, которая принимает дату в формате dd.mm.yyyy и определяет как сезон года, так и день недели. Программа должна учитывать особенности високосных годов и нестандартных календарей.

**3. Оснащение работы:** техническое задание, технический проект, ЭВМ, Kotlin, Itelij IDEA.

import java.time.LocalDate

import java.time.format.DateTimeFormatter

import java.time.format.DateTimeParseException

import java.util.Scanner

// Функция для проверки високосного года

fun isLeapYear(year: Int): Boolean {

return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)

}

fun main() {

val scanner = Scanner(System.in)

// Запрашиваем ввод даты

println("Введите дату в формате dd.MM.yyyy:")

val inputDate = scanner.nextLine()

// Используем DateTimeFormatter для парсинга даты

val formatter = DateTimeFormatter.ofPattern("dd.MM.yyyy")

try {

// Разбиваем строку даты на части вручную

val dateParts = inputDate.split(".")

if (dateParts.size != 3) {

throw DateTimeParseException("Неверный формат", inputDate, 0)

}

val day = dateParts[0].toInt()

val month = dateParts[1].toInt()

val year = dateParts[2].toInt()

// Проверка: если месяц февраль и день 29, проверяем високосный год

if (month == 2 && day == 29 && !isLeapYear(year)) {

println("29 февраля не существует в году $year, так как это не високосный год.")

} else {

// Проверка корректности даты с использованием LocalDate

val date = LocalDate.of(year, month, day)

// Дата валидна, продолжаем выполнение программы

val dayOfWeek = date.dayOfWeek.name.lowercase().replaceFirstChar { it.uppercase() }

val season = when (month) {

12 -> if (day >= 1) "Зима" else "Осень"

1, 2 -> "Зима"

3 -> if (day >= 1) "Весна" else "Зима"

4, 5 -> "Весна"

6 -> if (day >= 1) "Лето" else "Весна"

7, 8 -> "Лето"

9 -> if (day >= 1) "Осень" else "Лето"

10, 11 -> "Осень"

else -> "Неизвестно"

}

println("Дата: $inputDate")

println("День недели: $dayOfWeek")

println("Сезон: $season")

println("Год $year является високосным: ${isLeapYear(year)}")

}

} catch (e: DateTimeParseException) {

println("Неверный формат даты! Используйте формат dd.MM.yyyy")

} catch (e: Exception) {

println("Ошибка: ${e.message}. Проверьте правильность введенной даты.")

}

}

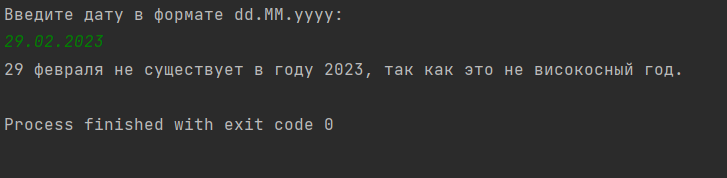


Рисунок 1 – Результат исполнения кода программы

**Вывод:** в ходе выполнения работы были изучены основные конструкции языка Kotlin, такие как типы данных, операции ввода-вывода, условные операторы, а также работа с переменными и операциями присваивания.