Министерство образования Республики Беларусь

Учреждение образования

Белорусский государственный университет

Информатики и радиоэлектроники

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Дисциплина: Программирование мобильных информационных систем

Отчет

по выполнению лабораторной работы №1

на тему

**«Основы языка Kotlin»**

Выполнил Медведюк А.А.  
ст. гр. 210101

Проверил преподаватель 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**

14 Определение следующей даты: Реализуйте программу, которая принимает дату и определяет следующую дату с учетом високосных годов, переходов между месяцами и годами.

**3 Оснащение работы и Программный код**

Техническое задание, технический проект, ЭВМ, Kotlin, Itelij IDEA.

fun main() {

while(true) {

println("Введите день:")

val day = readLine()?.toIntOrNull()

println("Введите месяц:")

val month = readLine()?.toIntOrNull()

println("Введите год:")

val year = readLine()?.toIntOrNull()

if (day != null && month != null && year != null && day <= 31 && month <= 12 && day > 0 && month > 0 && year >= 0) {

val (nextDay, nextMonth, nextYear) = getNextDate(day, month, year)

println("Текущая дата: $day.$month.$year")

println("Следующая дата: $nextDay.$nextMonth.$nextYear")

break

} else {

println("Invalid input")

}

}

}

fun getNextDate (day: Int, month: Int, year: Int): Triple<Int,Int,Int> {

val daysInMonth = getDaysInMonth(month,year)

var nextDay = day + 1

var nextMonth = month

var nextYear = year

if(nextDay > daysInMonth) {

nextDay = 1

nextMonth += 1

if(nextMonth > 12) {

nextMonth = 1

nextYear += 1

}

}

return Triple(nextDay,nextMonth,nextYear)

}

fun getDaysInMonth(month: Int, year: Int): Int {

return when(month) {

1, 3, 5, 7, 8, 10, 12 -> 31

4, 6, 9, 11 -> 30

2 -> if(isLeapYear(year)) 29 else 28

else -> throw IllegalArgumentException("Invalid month: $month")

}

}

fun isLeapYear(year: Int): Boolean {

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

}

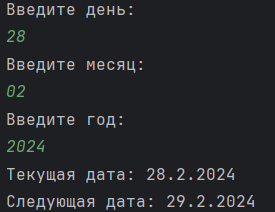


Рисунок 1 – Результат исполнения кода программы

**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.