Министерство образования Республики Беларусь

Учреждение высшего образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЕТ

по лабораторной работе №1

на тему

**Создание простых S-проектов на основе классов**

|  |  |
| --- | --- |
| Студенты группы 150501  Проверила | Михалович Т.В.  Петров В.В  Герман Ю. О. |

Минск 2023

**Цель:** изучить технику использования классов в Scala.

**Краткие теоретические сведения:**

Оbject используется для создания синглтон-объектов. Синглтон-объекты представляют собой уникальные экземпляры классов, которые создаются один раз и существуют в единственном экземпляре во всем приложении. Они часто используются для группировки функций и переменных, а также для создания статических членов класса. Синглтон-объекты в Scala обеспечивают удобный способ организации функционального кода и обеспечивают единственный точку доступа к его членам во всем приложении.

Основные характеристики синглтон-объектов в Scala:

* Единственный экземпляр: Синглтон-объект создается только один раз во время выполнения программы. Поэтому он обеспечивает гарантированную уникальность и доступность к своим членам.
* Ленивая инициализация: Синглтон-объекты инициализируются лениво, что означает, что они создаются только в момент первого обращения к ним.
* Доступ к членам: Вы можете определять методы и переменные внутри синглтон-объекта, и они могут быть вызваны и использованы с использованием имени объекта, как если бы это были статические члены класса.

В Scala функции могут быть определены с использованием ключевого слова *def*. Основные компоненты определения функции в Scala включают в себя: имя функции, должно быть уникальным в контексте, в котором она определена; параметры функции, перечисляются в круглых скобках после имени функции, и их типы указываются с помощью двоеточия; Тип возвращаемого значения, указывается после двоеточия, и он определяет тип данных, который функция будет возвращать; тело функции, содержит код, который выполняется при вызове функции.

Пример определения функции в Scala:

def add(x: Int, y: Int): Int = {

return x + y

}

**Выполнение работы:**

Перед выполнением основного задания лабораторной работы мы выполнили рекомендованные к изучению примеры.

В первом примере мы познакомились с базовыми основами языка программирования Scala, такие как операции с классами и объектами. Результатом исполнения программы является нижеприведенный скриншот:

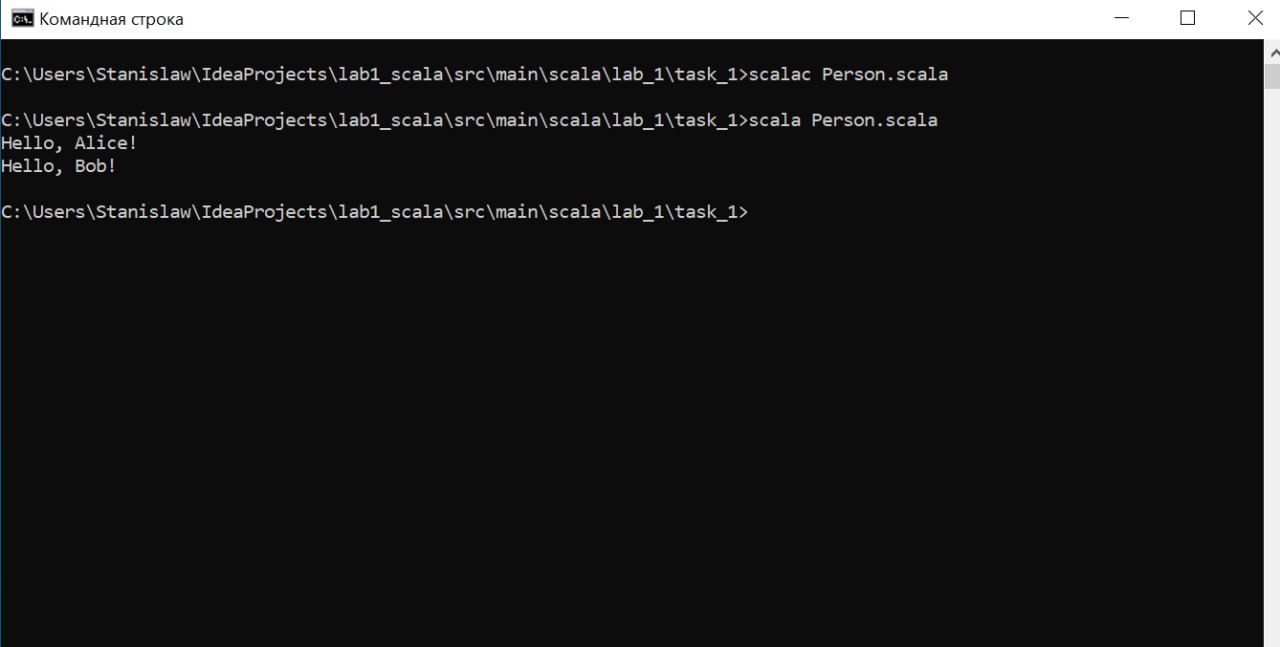


Рисунок 1 — Результат выполнения первого примера

Во втором примере с помощью языка Scala находили НОД двух чисел, вводимых с клавиатуры. Результатом исполнения программы является нижеприведенный скриншот:

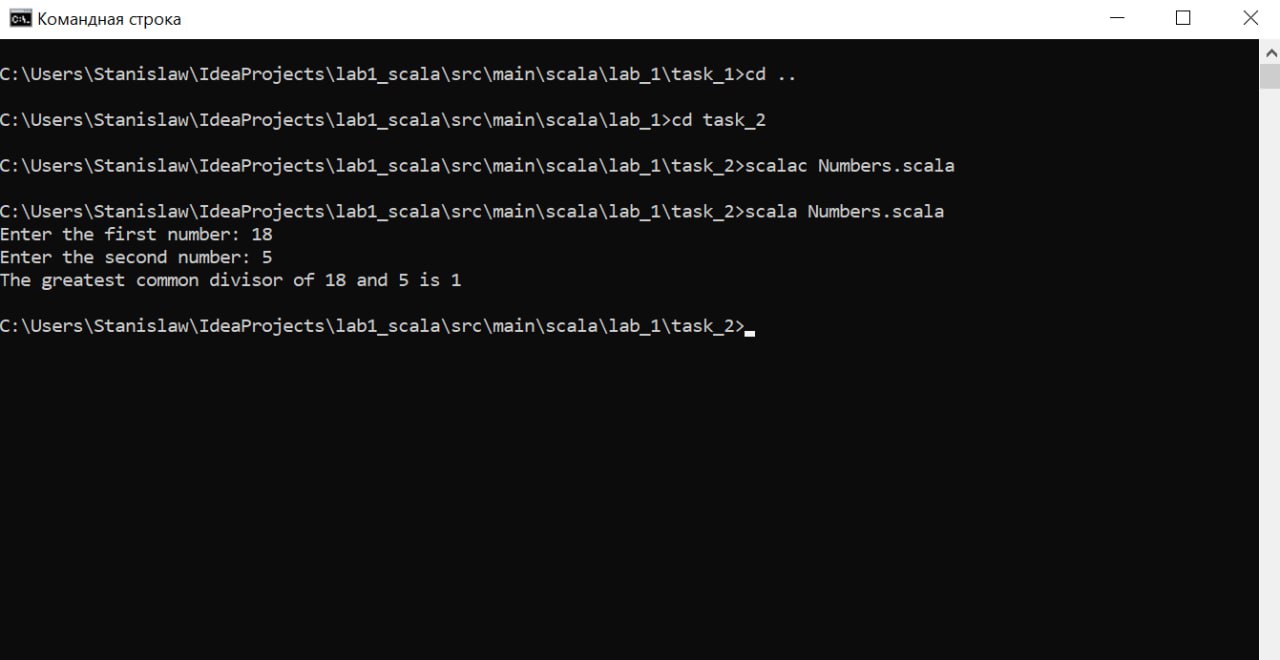


Рисунок 2 — Результат выполнения второго примера

В третьем примере с помощью языка Scala формировали целое число из первых трех чисел вводимого шестизначного числа. Результатом исполнения программы является нижеприведенный скриншот:

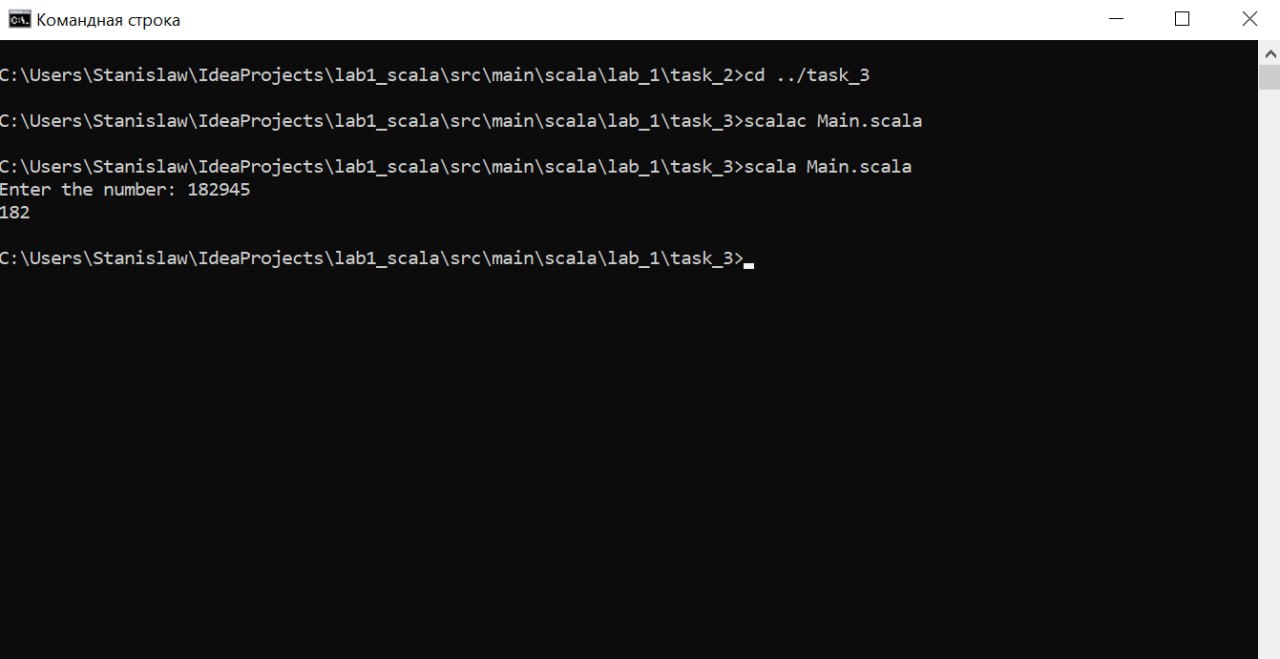


Рисунок 3 — Результат выполнения третьего примера

Так же нам было порекомендовано проанализировать код и запустить один из вариантов лабораторной работы. В данном варианте было поручено разработать генератор случайных чисел. Результатом исполнения данной программы являются скриншоты, приведенные ниже.

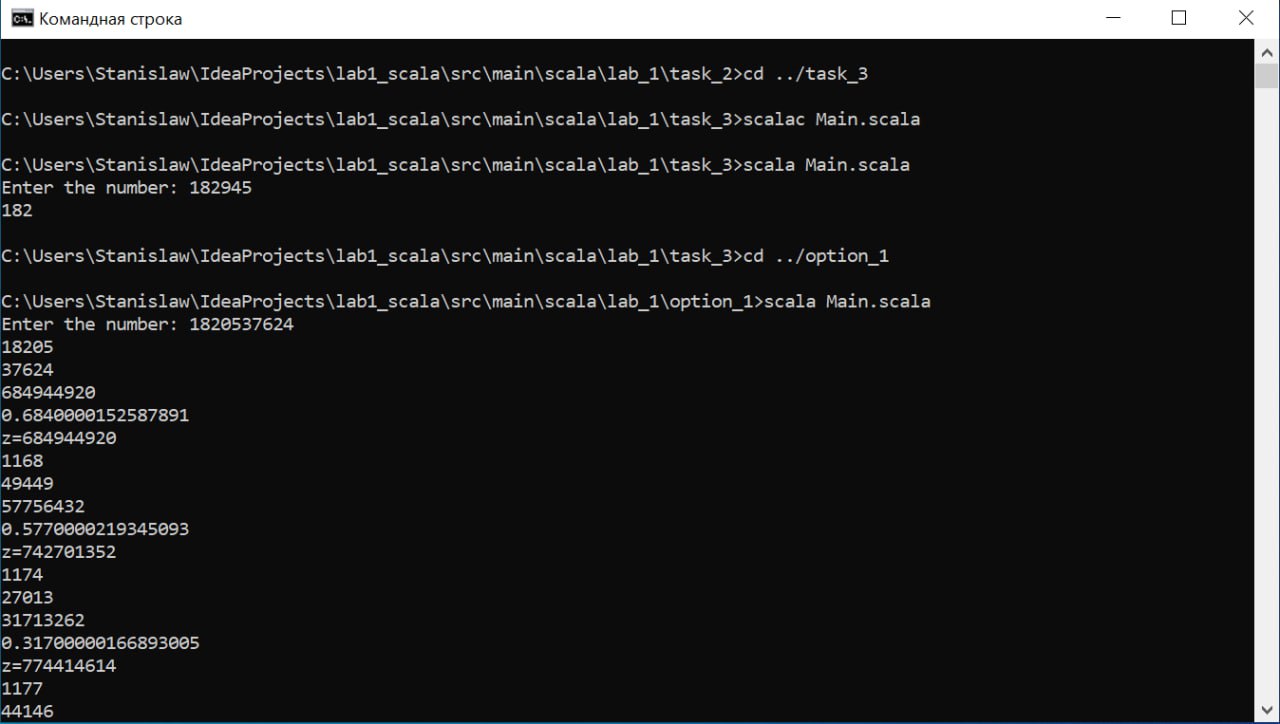


Рисунок 4 – Результат выполнения первого варианта (1 часть)

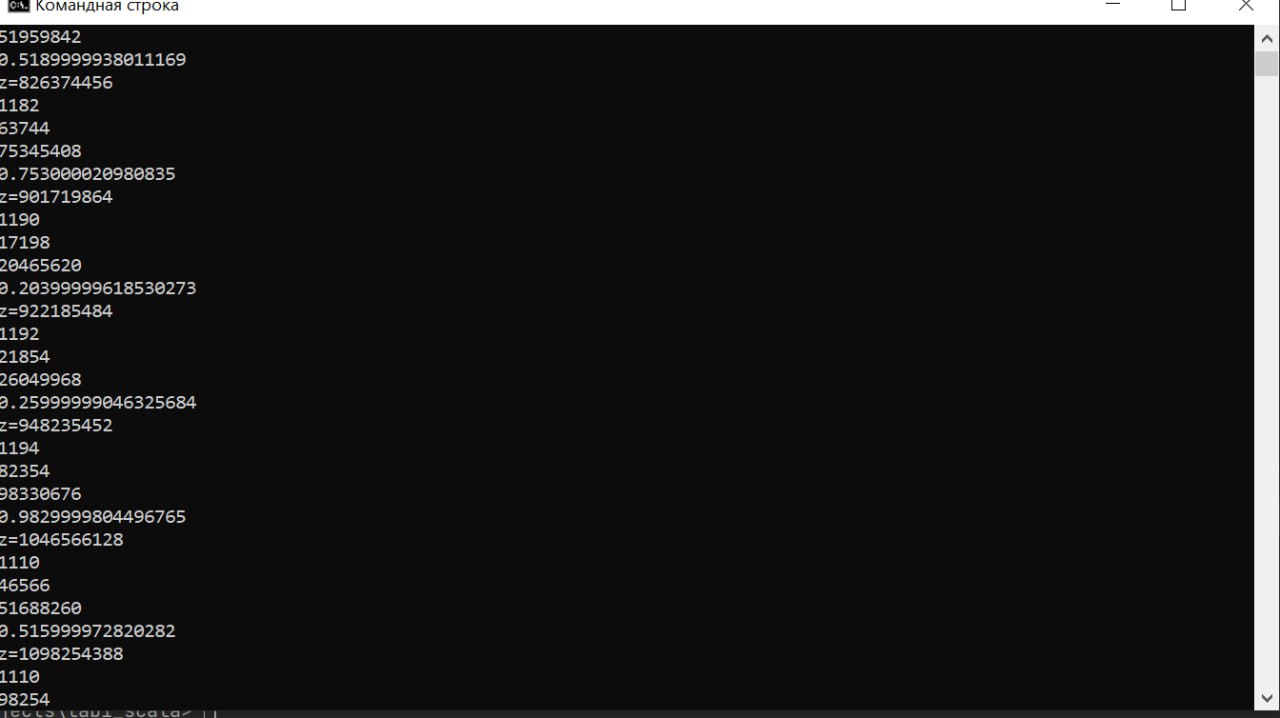


Рисунок 5 – Результат выполнения первого варианта (2 часть)

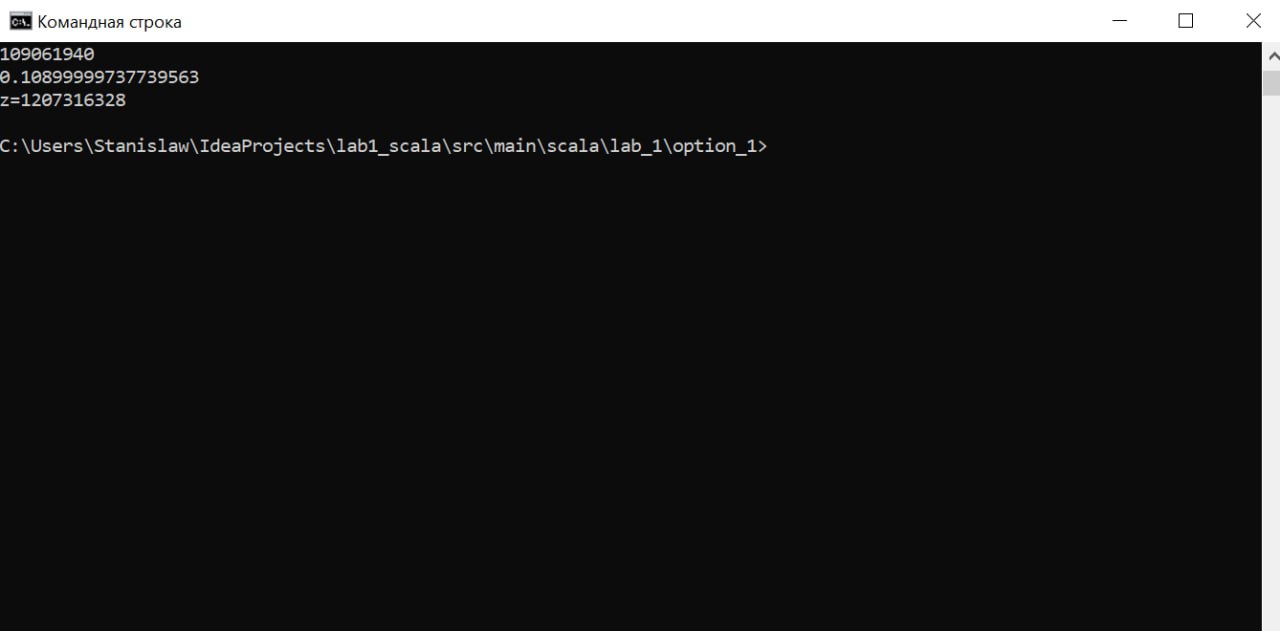


Рисунок 6 – Результат выполнения первого варианта (3 часть)

На основе первого варианта лабораторной работы было произведено выполнения второго варианта:

Задание: Использовать предыдущий пример, но только число А получается из цифр, стоящих на четных позициях, а В - на нечетных.

Код программы содержит один scala-объекта: и Main. Объект Main имеет точку входа в приложение, в нем соответственно выполняются методы substringFromSymbolsAtEvenPositions и substringFromSymbolsAtOddPositions.

1. Метод substringFromSymbolsAtEvenPositions(): В данном методе из числа, переданного в параметрах метода. Формируется новое число и цифр параметра, стоящих на четных позициях.

2. Метод substringFromSymbolsAtOddPositions(): В данном методе из числа, переданного в параметрах метода. Формируется новое число и цифр параметра, стоящих на нечетных позициях.

Листинг кода программы:

Результаты выполнения задания представлены скриншотами ниже:

object Main234 {

def main(args: Array[String]): Unit = {

print("Enter the number: ")

var numberString =""

var a =0

var b=0

var c=0

var rez: Double =0.0

var z=0

numberString = scala.io.StdIn.readLine()

if ( numberString.length() != 10) {

println("Incorrect number");

sys.exit(0) }

for (i <- 1 to 5) {

a = substringFromSymbolsAtEvenPositions(numberString).toInt

b= substringFromSymbolsAtOddPositions(numberString).toInt

c= a\*b

rez=c.toString.substring(0,3).toFloat / 1000

z+=c

if(z < 0){

z = z.abs

numberString = "011" + z.toString

}

if (z.toString.length() < 10){

numberString = z.toString

for (j <- 1 to (10 - numberString.length() + 1)){

numberString = "1" + numberString

}

numberString = "0" + numberString

println("new numberString = " + numberString)

}

else {

numberString = z.toString

}

println("a = " + a)

println("b = " + b)

println("c = " + c)

println("rez = " + rez)

println("z ="+ z)

if(numberString.length < 12){

numberString = "11" + numberString

}

numberString= numberString.substring(1,numberString.length()-1)

println("numberString = " + numberString)

println()

}

}

def substringFromSymbolsAtEvenPositions(str: String): String = {

val result = new StringBuilder

for (i <- str.indices if i % 2 == 1) {

result.append(str(i))

}

result.toString()

}

def substringFromSymbolsAtOddPositions(str: String): String = {

val result = new StringBuilder

for (i <- str.indices if i % 2 != 1) {

result.append(str(i))

}

result.toString()

}

}

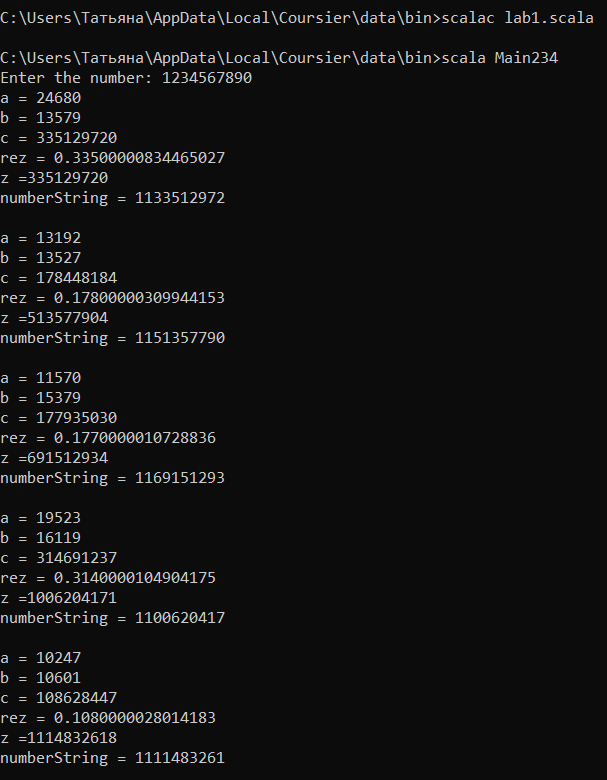


Рисунок 7 – Результат выполнения второго варианта

Вывод: В ходе выполнения данной лабораторной работы нами были проанализированы и изучены техники использования классов в Scala.