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 150503:  Проверила: | Ходосевич М.А.  Семков А. Д.  Герман Ю. О. |

Минск 2023

Цель: изучить технику использования классов в Scala.

1. **Краткие теоретические сведения:**

Scala приложение – это набор объектов, которые связаны между собой вызовом методов друг друга.  Основные сущности данного языка программирования:

* **Объект -** объект является единственным экземпляром своего класса.
* **Класс -** класс является шаблоном, описывающим состояние (поля) и поведение (методы) объектов, созданных на его основе.
* **Метод -** метод представляет собой блок кода, который выполняет определенную задачу.
* **Поля -** представляют собой переменные, которые хранят данные объекта.

Пример определения функции в Scala:

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

return x + y

}

Выполнение рекомендованных примеров к данной лабораторной работе:

class Person {

def hello(name :String ): Unit = println(s"hello, $name")

object Main22 {

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

val x = new Person();

x.hello("matvey");

x.hello("man");

Здесь объявлен класс Person с единственным методом (функцией) hello. Аргументом функции является строковая переменная name. Функция выводит это значение на экране в операторе

Пример выполнения данного кода приведен на рисунке 1.

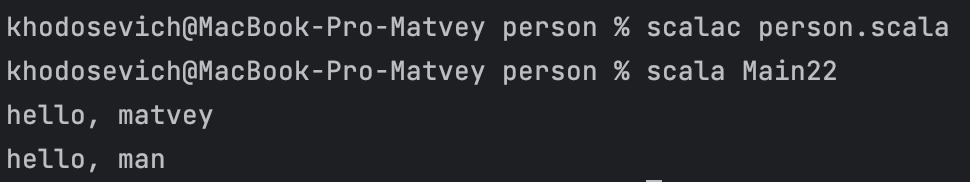


рисунок 1 - пример выполнения примера

Во втором примере нужно ввести с клавиатуры два целых числа и найти их наибольший целый (общий) делитель

object Numbers {

def gcd(a: Int, b: Int): Int = {

if (b == 0) a else gcd(b, a % b)

}

}

object Main22 {

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

print("Enter the first number: ")

val a = scala.io.StdIn.readInt()

print("Enter the second number: ")

val b = scala.io.StdIn.readInt()

val gcdValue = Numbers.gcd(a, b)

println(s"The greatest common divisor of $a and $b is $gcdValue")

}

Пример выполнения данного кода приведен на рисунке 2.

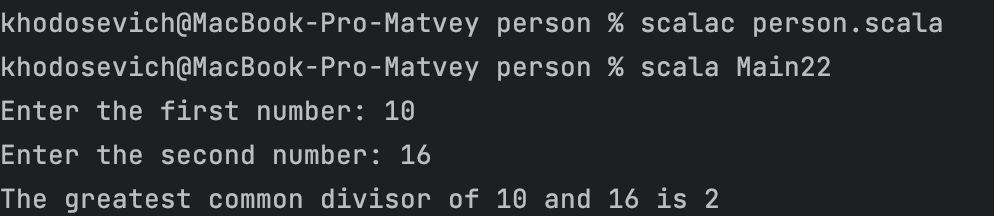


рисунок 2 - пример выполнения примера

В третьем примере сформировать целое число из первых трех цифр шестизначного числа:

object Main22 {

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

print("Enter the number: ")

val number = scala.io.StdIn.readInt

try {

val numberString = number.toString

val firstThreeDigits = if (numberString.length >= 3 && number >= 0) {

numberString.substring(0, 3).toInt

} else {

throw new Exception("Number does not have at least three digits or is negative")

}

println(firstThreeDigits)

} catch {

case e: Exception => println(e.getMessage)}}}

Пример выполнения данного кода приведен на рисунке 3.

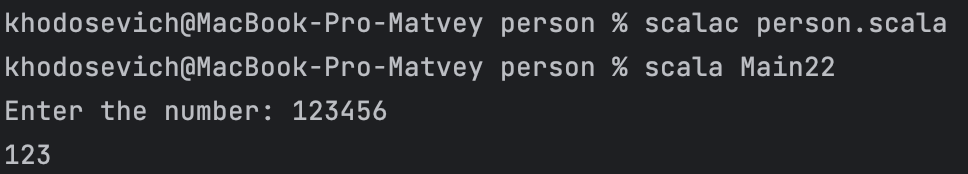


рисунок 3 - пример выполнения примера

**2. Ход работы:**

**Вариант 1.**

Построить генератор случайных чисел по следующей схеме. Вводите 10-значное число с клавиатуры. Формируете два новых целых числа A и B: первое A состоит из первых пяти цифр, второе B – из последних цифр введенного числа, начиная с шестой. Перемножаете числа A и B друг на друга. Первые три цифры результата С, будучи поделенными на 1000, дают первое случайное число REZ. Чтобы сформировать следующее случайное число, прибавляете к Z число С и повторяете процесс. Формируете два новых целых числа A и B: первое A состоит из первых пяти цифр, второе B – из последних цифр числа Z, начиная с шестой. Перемножаете числа A и B друг на друга. Первые три цифры результата С, будучи поделенными на 1000, дают второе случайное число. Чтобы сформировать следующее случайное число, прибавляете к Z число С и повторяете процесс. Сформируйте 5 случайных чисел.

Если z отрицателен, то это значит, что надо заменить первую цифру единицы на ноль и убрать знак “минус”, кроме того, нужно поддерживать длину строки не меньше 10 символов.

Исходный код работы программы первого варианта:

object Variant1 {

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 10) {

a = numberString.substring(0, 5).toInt

b= numberString.substring(5, 10).toInt

c= a\*b

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

z+=c

println(s"\na = ${a} b = ${b} c = ${c} rez = ${rez} z = ${z}" );

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

numberString="011" + z}}}

Пример выполнения данного варианта приведен на рисунке 4.

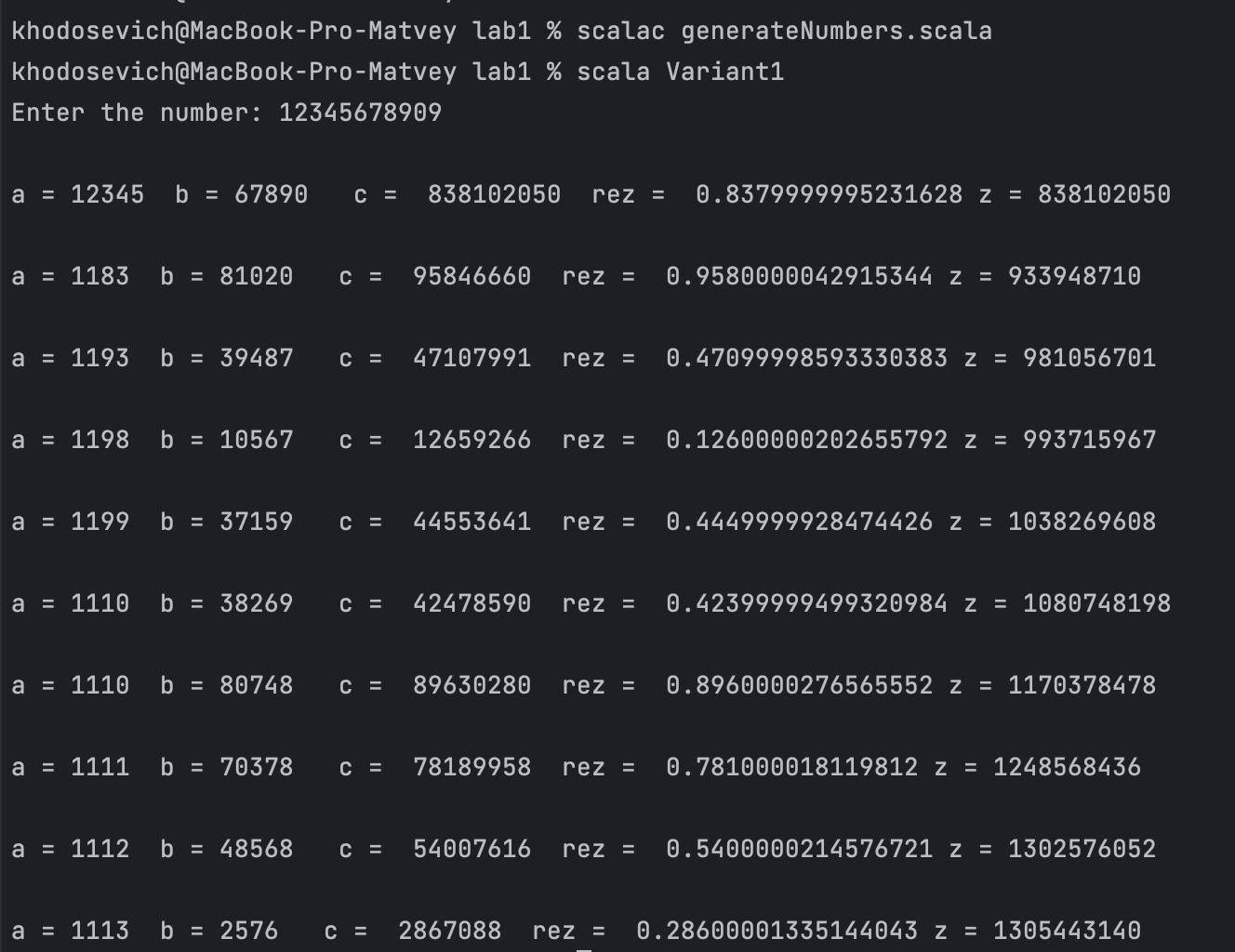
****

рисунок 4 - пример выполнения первого варианта

**Вариант 2.**

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

Сформируйте по этой схеме 5 случайных чисел.

Исходный код работы программы первого варианта:

object Variant2 {

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 == 0) {

result.append(str(i))

}

result.toString()

}

def inputNumber() : String = {

var numberStr : String = "";

print ("Enter the number: ")

numberStr = scala.io.StdIn.readLine ()

if ((numberStr.length () < 10) ) {

println ("Incorrect number");

sys.exit (0)

} else {

numberStr.substring (0, 10)

}

numberStr

}

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

var a: Long = 0

var b : Long = 0

var c : Long = 0

var rez: Float = 0

var z : Long = 0

var numberString: String = inputNumber();

for (i <- 1 to 5) {

a = substringFromSymbolsAtEvenPositions(numberString).toLong

b = substringFromSymbolsAtOddPositions(numberString).toLong

c = a \* b

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

z += c

if(z < 0 ){

z \*= -1;

}

if(z.toString.length > 11) z = z.toString.substring(0,10).toLong;

println(s"\na = ${a} b = ${b} c = ${c} rez = ${rez} z = ${z}" );

numberString = "011" + z

numberString = numberString.substring(0, 11)

}}}

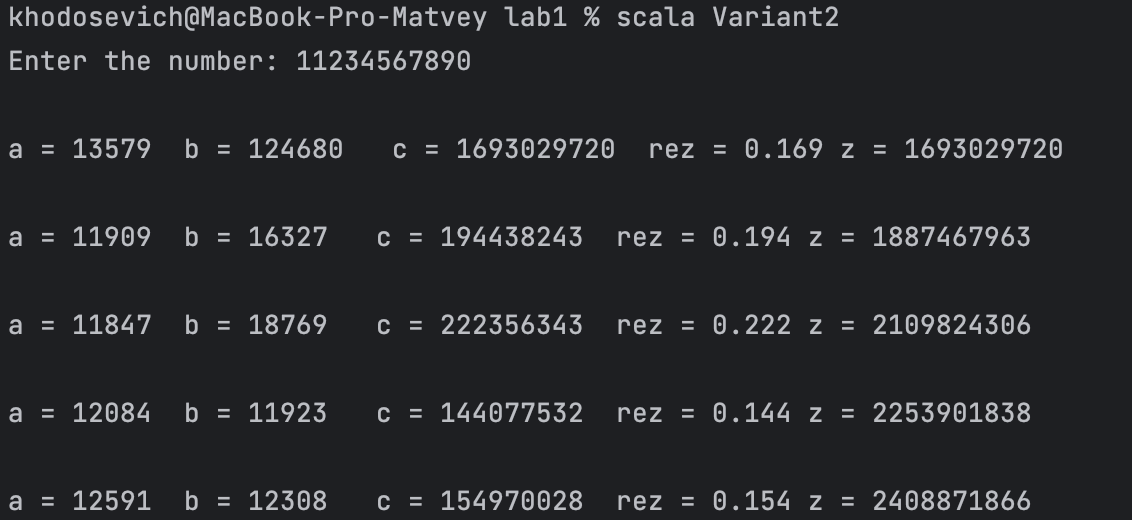
Пример выполнения данного варианта приведен на рисунке 5.

рисунок 5 - пример выполнения второго варианта

1. **Вывод:**

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