ЛАБОРАТОРНАЯ РАБОТА 1. Создание простых S-проектов на основе классов

Scalac person.scala

Scala Main22

Цель. Изучить технику использования классов в Scala.

Краткие теоретические сведения.

Напишем следующий пример с классом в текстовом файле lab4.scala

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("Alice")

x.hello("Bob")

}

}

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

println(s"Hello, $name!")

Не забываем перед телом функции ставить знак равно

= println(s"Hello, $name!")

Хорошим стилем будет вообще заключить тело функции в фигурные скобки, да еще поставить точку с запятой в конце оператора:

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("Alice")

x.hello("Bob")

}

}

Заметим, что в операторе

println(s"Hello, $name!");

терм $name используется для подстановки значения переменной name (предшествует символ доллара).

Теперь мы хотим 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")

}

}

Читаем целые числа с клавиатуры таким образом

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

Сделаем модификации. Вводим числа как строки и конвертируем в числа:

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.readLine().toInt

print("Enter the second number: ")

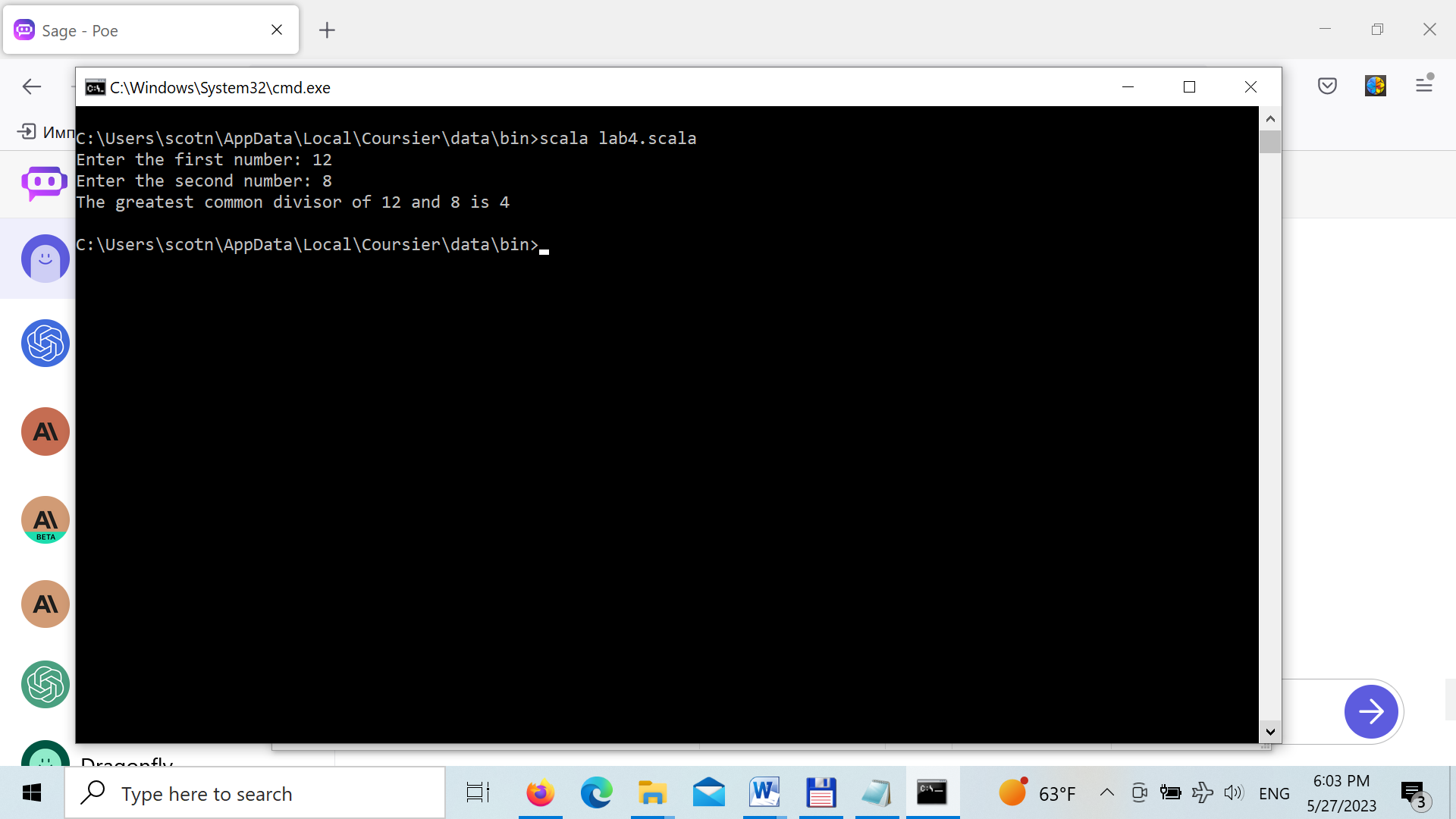
val b = scala.io.StdIn.readLine().toInt

val gcdValue = Numbers.gcd(a, b)

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

}

}



Определить сумму цифр в записи целого числа

object Main22 {

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

print("Enter the number: ")

val a = scala.io.StdIn.readInt

var sum = 0

for (digit <- a.toString) {

sum += digit.asDigit

}

// Print the sum

println(sum)

}

}

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)

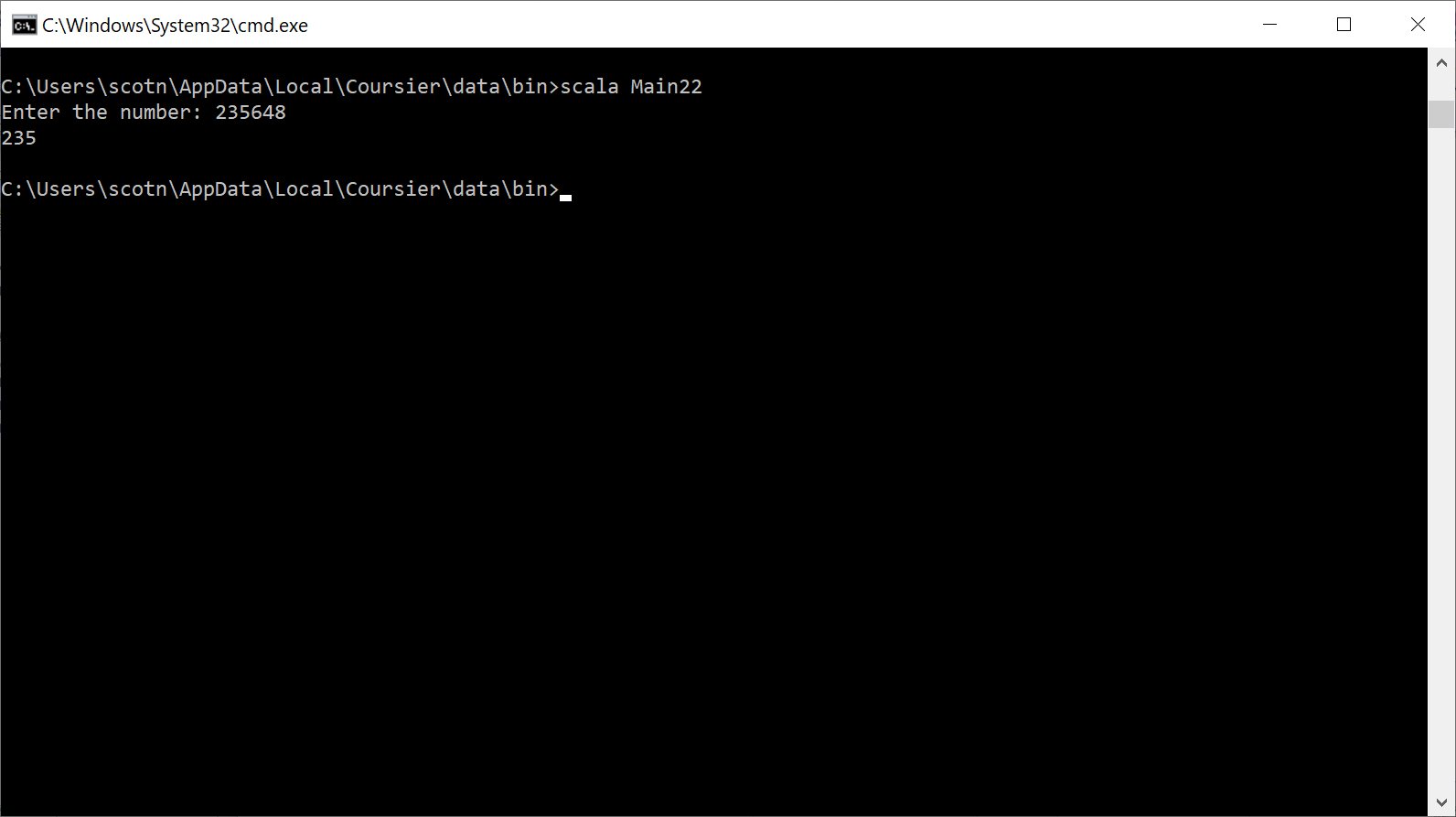
} catch {

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

}

}

}



Записать число в обратном порядке цифр:

// Write a number in the reversal order of its digits

val number = 12345

val reversedNumber = number.toString.reverse.toInt

// Print the reversed number

println(reversedNumber)

***Порядок выполнения работы***

1. Изучить теоретическую часть.
2. Получить индивидуальное задание у преподавателя из прилагаемого списка.
3. Создать и отладить приложение в среде Python.
4. Написать отчет.
5. Защитить работу.

***Индивидуальные задания***

**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. Сказанное, демонстрируется следующим примером:

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: Float =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

//println(numberString.length());

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

c= a\*b

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

z+=c

println(a);

println(b);

println(c);

println(rez);

println("z="+ z)

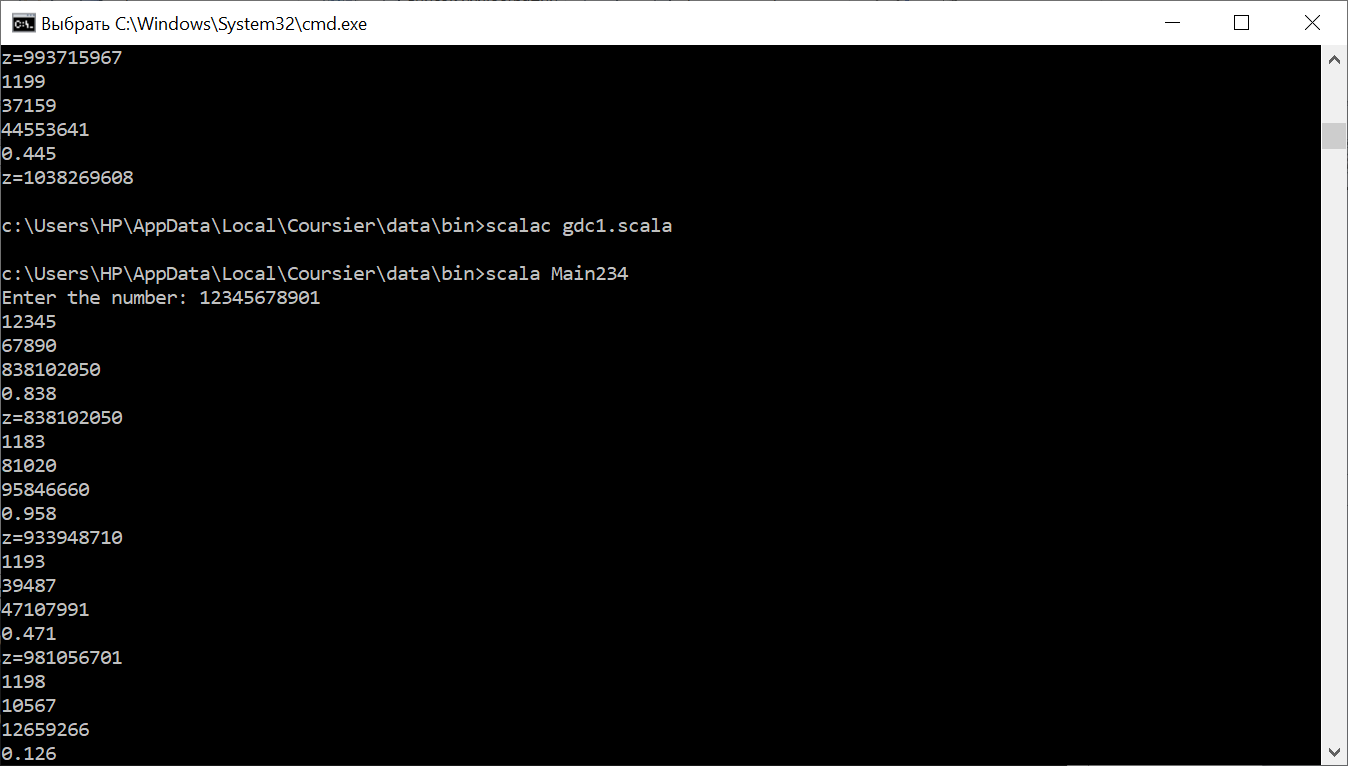
numberString= numberString.substring(1,numberString.length()-1)

numberString="011"+z

}

}

}



**Вариант 2.**

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

def substringFromSymbolsAtEvenPositions(str: String): String = {

val result = new StringBuilder

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

result.append(str(i))

}

result.toString()

}

object Main234 {

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

val originalString = "1234567890"

val substring = substringFromSymbolsAtEvenPositions(originalString)

println(substring)

}

}

Сформируйте по этой схеме 5 случайных чисел.

**Вариант 3.**

Построить генератор случайных чисел по следующей схеме. Вводите 10-значное число Z с клавиатуры. 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. Дописываете к числу В число А. Получаете число W. Находите С= Z\*W. Первые три цифры результата С, будучи поделенными на 1000, дают первое случайное число REZ. Чтобы сформировать следующее случайное число, формируете два новых целых числа A и B: первое A состоит из первых пяти цифр, второе B – из последних цифр числа C, начиная с шестой. Дописываете к числу В число А. Получаете число W. Находите C= C\*W. Первые три цифры результата C, будучи поделенными на 1000, дают второе случайное число REZ. Сформируйте 3 случайных чисел по этой схеме.

**Вариант 4.**

Построить генератор случайных чисел по следующей схеме. Вводите 16-значное число Z с клавиатуры. Формируете два новых целых числа A и B: первое A состоит из первых восьми цифр, второе B – из последних цифр числа Z, начиная с девятой. Перемножаете числа A и B друг на друга, получаем число С. Первые три цифры результата С, будучи поделенными на 1000, дают первое случайное число REZ. Чтобы сформировать следующее случайное число, умножаете Z на число С и оставляете 16 цифр в результате. Повторяете процесс для сформированного таким образом числа Z. Формируете два новых целых числа A и B: первое A состоит из первых восьми цифр, второе B – из последних цифр числа Z, начиная с девятой. 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, умножаете Z на число С и повторяете процесс. Сформируйте 5 случайных чисел.

***Контрольные вопросы***

1. Что такое object?
2. Как описываются функции в SCALA?
3. Как использовать класс в Object?
4. Как определяется тип возвращаемого функцией значения?