

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И
МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Ордена Трудового Красного Знамени федеральное государственное
бюджетное образовательное учреждение высшего образования
«Московский технический университет связи и информатики»**

Лабораторная работа №1

по дисциплине:

«Введение в информационные технологии»

Выполнил:

Студент группы БФИ19

Соколовский Н.Р

Москва, 2021

Задание на лабораторную работу

1. Переменные `res` – это значения `val` или настоящие переменные `var` (рисунок 1)?

`Val`, так как при попытке `res0=5` повторяется сообщение об ошибке "переназначение на `val`" => переменные `res`-это `val`.

```
scala> res0=5
-- Error:
1 | res0=5
  | ^^^^
  | Not found: res0

scala> val res0=5
val res0: Int = 5

scala>
```

Рисунок 1 – Задание 1

Значение, объявленное с помощью `val`, в действительности является константой – ее значение нельзя изменить. Чтобы объявить переменную, значение которой может изменяться, следует использовать ключевое слово `var`. В языке Scala предпочтительнее использовать `val`, если в дальнейшем не предполагается изменять значение.

2. `"crazy" * 3` в REPL (рисунок 2).

```
"crazy" * 3
val res0: String = crazycrazycrazy

scala>
```

Рисунок 2 – Задание 2

3. Что означает выражение `10 max 2`? В каком классе определен метод `max` (рисунок 3)?

Метод `max` выбирает наибольшее из пары чисел и его возвращает. Определен для классов `Int`, `Long`, `Float`, `Double`.

```
scala> 10 max 2
val res1: Int = 10

scala>
```

Рисунок 3 – Задание 3

4. Используя число типа BigInt, вычислите 2^{1024} (рисунок 4).

```
scala> BigInt(2).pow(1024)
val res2: BigInt = 17976931348623159077293051907890247336179769789423065727343008115773267580
55009631327084773224075360211201138798713933576587897688144166224928474306394741243776789342
486548527630221960124609411945308295208500576883815068234246288147391311054082723716335051068
4586298239947245938479716304835356329624224137216

scala>
```

Рисунок 4 – Задание 4

5. Что нужно импортировать, чтобы найти случайное простое число вызовом метода `probablePrime(100, Random)` без использования каких-либо префиксов перед именами `probablePrime` и `Random` (рисунок 5)?

```
scala> import scala.BigInt.probablePrime
scala> import scala.util.Random
scala> probablePrime(100, Random)
val res3: BigInt = 799171283703294937037535595667

scala>
```

Рисунок 5 – Задание 5

6. Один из способов создать файл или каталог со случайным именем состоит в том, чтобы сгенерировать случайное число типа `BigInt` и преобразовать его в систему счисления по основанию 36, в результате получится строка, такая как "qsnvbevtohcj38o06kul". Отыщите в Scaladoc методы, которые можно было бы использовать для этого (рисунок 6). Нужно сгенерировать строку. Используем `BigInt`. Есть метод `toString`, куда мы передаем аргумент (систему счисления). И строка генерируется.

```
Command Prompt - scala3-repl

val res2: BigInt = 17976931348623159077293051907890247336179769789423065727343008115773
267580550096313270847732240753602112011387987139335765878976881441662249284743063947412
437776789342486548527630221960124609411945308295208500576883815068234246288147391311054
0827237163350510684586298239947245938479716304835356329624224137216

scala> import scala.BigInt.probablePrime

scala> import scala.util.Random

scala> probablePrime(100, Random)
val res3: BigInt = 799171283703294937037535595667

scala> probablePrime(100, Random).toString(36)
val res4: String = 1yo4gb2y8y75bj15y40h

scala>
```

Рисунок 6 – Задание 6

7. Как получить первый символ строки в языке Scala? А последний символ (рисунок 7)?

```
Command Prompt - scala3-repl

scala> import scala.util.Random

scala> probablePrime(100, Random)
val res3: BigInt = 799171283703294937037535595667

scala> probablePrime(100, Random).toString(36)
val res4: String = 1yo4gb2y8y75bj15y40h

scala> "Slovo".head
val res5: Char = S

scala> "Slovo".last
val res6: Char = o

scala>
```

Рисунок 7 – Задание 7

8. Что делают строковые функции `take`, `drop`, `takeRight` и `dropRight`? Какие преимущества и недостатки они имеют в сравнении с `substring` (рисунок 8)?

`take(n)` – берет первые `n` символов,

`drop(n)` – удаляет первые `n` символов,

`takeRight(n)` – берет справа первые `n` символов,

`dropRight(n)` – удаляет справа первые `n` символов.

```
Command Prompt - scala3-repl
scala> "Slovo".last
val res6: Char = o

scala> "Slovo".take(2)
val res7: String = Sl

scala> "Slovo".drop(2)
val res8: String = ovo

scala> "Slovo".takeRight(2)
val res9: String = vo

scala> "Slovo".dropRight(2)
val res10: String = Slo

scala>
```

Рисунок 8 – Задание 8

```
scala> "Slovo".substring(2)
val res11: String = ovo

scala>
```

9. Сигнум числа равен 1, если число положительное. -1 – если отрицательное, и 0 – если равно нулю. Напишите функцию, вычисляющую это значение (рисунок 9).

```
scala> def signum(n: Int): Int = {
  |   if (n>0) 1 else if (n<0) -1 else 0
  | }
def signum(n: Int): Int

scala> signum(4) == 1
val res12: Boolean = true

scala> signum(8) == 1
val res13: Boolean = true

scala> signum(-8) == 1
val res14: Boolean = false

scala> signum(0) == 1
val res15: Boolean = false

scala>
```

Рисунок 9 – Задание 9

10. Какое значение возвращает блок {}? Каков его тип?
Unit

11. Напишите на языке Scala цикл, эквивалентный циклу на языке Java (рисунок 10).

for (int i=10; i>=0; i--) System.out.println(i);

```
scala>for (i<-10.to(0, -1))println(i)
10
9
8
7
6
5
4
3
2
1
0

scala>
```

Рисунок 10 – Задание 11

12. Напишите процедуру countdown (n: Int), которая выводит числа от n до 0 (рисунок 11).

```
scala> def countdown(n: Int): Unit = {
|   for (i<-n.to(0, -1)) println(i)
| }
def countdown(n: Int): Unit

scala> countdown(10)
10
9
8
7
6
5
4
3
2
1
0

scala>
```

Рисунок 11 – Задание 12

13. Напишите цикл for для вычисления кодовых пунктов Юникода всех букв в строке. Например, произведение символов в строке «Hello» равно 9415087488L (рисунок 12).

```
scala> var a: Long = 1
var a: Long = 1

scala> for (x <- "Hello") a *=x.toLong

scala> a
val res16: Long = 9415087488

scala>
```

Рисунок 12 – Задание 13

14. Решите предыдущее упражнение без применения цикла. Напишите функцию `product(s: String)`, вычисляющую произведение, как описано в предыдущих упражнениях.

```
scala> def product(s: String): Long = {
  |   return s.map(_._toLong).product
  | }
def product(s: String): Long

scala> product("Hello")
val res17: Long = 9415087488

scala>
```

Рисунок 13 – Задание 14

16. Сделайте функцию из предыдущего упражнения рекурсивной (рисунок 15).

```
scala> def product(s: String): Long = {
  |   if (s.tail != "") s.head.toLong * product(s.tail) else s.head.toLong
  | }
def product(s: String): Long

scala> product("Hello")
val res18: Long = 9415087488

scala>
```

Рисунок 15 – Задание 16

17. Напишите функцию, вычисляющую x_n , где n – целое число. Используйте следующее рекурсивное определение:

- $x_n = y^2$, если n – четное и положительное число, где $y = x_{n/2}$
- $x_n = x * x_{n-1}$, если n – нечетное и положительное число.
- $x_0 = 1$.
- $x_n = 1/x_{-n}$, если n – отрицательное число.

Не используйте инструкцию `return`.

```
Command Prompt - scala3-repl

scala> import scala.math.pow

scala> def compute(x: Double, n: Int): Double = {
  | if (n>0 && n%2 == 0) pow(pow(x,n/2),2)
  | else
  | if (n>0 && n%2 != 0) x*pow(x,n-1)
  | else
  | if (n<0)1/pow(x,-n)
  | else
  | 1
  | }
def compute(x: Double, n: Int): Double

scala> compute(5,0)
val res19: Double = 1.0

scala> compute(11,3)
val res20: Double = 1331.0

scala> compute(15,5)
val res21: Double = 759375.0

scala>
```

Рисунок 16 – Задание 17

18. $f(m,n)$ - сумма всех натуральных чисел от m до n включительно, в десятичной записи которых нет одинаковых цифр.

```
scala> def f(m: Int, n: Int): Int = {
  | return Range(m, n+1).filter(num=>num.toString().distinct == num.toString()).sum
  | }
def f(m: Int, n: Int): Int

scala> f(14,12)
val res22: Int = 0

scala> f(12,14)
val res23: Int = 39

scala>
```

Рисунок 17 – Задание 18

19. Список содержит целые числа, а также другие списки, такие же как и первоначальный. Получить список, содержащий только целые числа из всех вложенных списков.

Пример:

$f(\text{List}(\text{List}(1, 1), 2, \text{List}(3, \text{List}(5, 8)))) = \text{List}(1, 1, 2, 3, 5, 8)$


```
scala> def f(list: List[Int | List[Int]]): List[Int] = {
    |   var result: List[Int] = List()
    |   list foreach {
    |     case item: List[Int] => result = result ++ item
    |     case item: Int => result = result :+ item
    |   }
    |   return result
    | }
def f(list: List[Int | List[Int]]): List[Int]

scala> f(List(1,2,3,List(4,5,6),7))
val res7: List[Int] = List(1, 2, 3, 4, 5, 6, 7)

scala>
```

Рисунок 18 – Задание 19

20. $f(n)$ - сумма цифр наибольшего простого делителя натурального числа n .

```
scala> def f( n: Int): Int = {
    |   return Range(1,n+1)
    |   .filter(num=>(n%num == 0) && Range(2,num).find(divisor => num % divisor == 0).isEmpty).max.toString.map(_.asDigit).sum}
def f(n: Int): Int

scala> f(6)
val res26: Int = 3

scala> f(12)
val res27: Int = 3

scala>
```

Рисунок 19 – Задание 20

21. Список содержит элементы одного, но любого типа. Получить список, содержащий каждый имеющийся элемент старого списка k раз подряд. Число k задается при выполнении программы.

```
val c: Int = 3

scala> val list: List[Any] = List(7,6,5)
val list: List[Any] = List(7, 6, 5)

scala> var result: List[Any] = List()
var result: List[Any] = List()

scala> list.foreach(value => Range(1, c+1).foreach((_) => result = result:+value))

scala> println(result)
List(7, 7, 7, 6, 6, 6, 5, 5, 5)
```

Рисунок 20 – Задание 21

24. $f(m,n)$ - наименьшее общее кратное натуральных чисел m и n .

```
scala> def f(m: Int, n: Int): Int = {  
    | Range(m max n, Int.MaxValue).find(num => num%n == 0 && num%m == 0)  
    | match {case Some(value) => value  
    |       case None => -1  
    | }  
    | }  
def f(m: Int, n: Int): Int  
  
scala> f(10)  
-- Error:  
1 | f(10)  
  | ^^^^^  
  | missing argument for parameter n of method f: (m: Int, n: Int): Int  
  
scala> f(10,21)  
val res28: Int = 210  
  
scala>
```

Рисунок 21 – Задание 24

25. Список содержит элементы одного, но любого типа. Получить список, из элементов исходного, удаляя каждый k -й элемент. Число k задается при выполнении программы.

```
scala> val k=3  
val k: Int = 3  
  
scala> val list: List[Any] = List(1,2,3,1,2,3,1,2,3)  
val list: List[Any] = List(1, 2, 3, 1, 2, 3, 1, 2, 3)  
  
scala> var result: List[Any]=List()  
var result: List[Any] = List()  
  
scala> list.zip(1 until list.size+1).foreach(tuple => if(tuple._2%k!=0)(result=result :+ tuple._1))  
  
scala> result  
val res1: List[Any] = List(1, 2, 1, 2, 1, 2)  
  
scala>
```

Рисунок 22 – Задание 25

26. $f(n,k)$ - число размещений из n по k . Факториал не использовать.

```
scala> def factorial(n : Int): Int = {
    |   if(n==1) return 1
    |   return n*factorial(n-1)
    | }
def factorial(n: Int): Int

scala> def fun(n: Int, k: Int): Double = {
    |   return factorial(n) / factorial(n-k)
    | }
def fun(n: Int, k: Int): Double

scala> fun(8,4)
val res2: Double = 1680.0

scala>
```

Рисунок 23 – Задание 26

27. Список содержит элементы одного, но любого типа. Получить новый список, перемещая циклически каждый элемент на k позиций влево (при перемещении на одну позицию первый элемент становится последним, второй первым и так далее). Число k задается при выполнении программы.

```
scala> def f (list: List[Any], k: Int) : List[Any] = {
    |   var result: List[Any]= List()
    |   if (k==0) return result
    |   if (k>0) {
    |     for(i<-k to list.length -1){
    |       result = result :+ list(i)
    |     }
    |     for (j<-0 to k-1){
    |       result = result :+ list(j)
    |     }
    |   } else {
    |     var m = -k
    |     for (i<-0 to list.length - m +1){
    |       result = result :+ list(i)
    |     }
    |     for ( j<- list.length - 1 to list.length - m by -1){
    |       result = list(j) :: result
    |     }
    |   }
    |   return result
    | }
def f(list: List[Any], k: Int): List[Any]

scala> f(List(1,2,3,4,5,6,7,8),3)
val res3: List[Any] = List(4, 5, 6, 7, 8, 1, 2, 3)

scala>
```

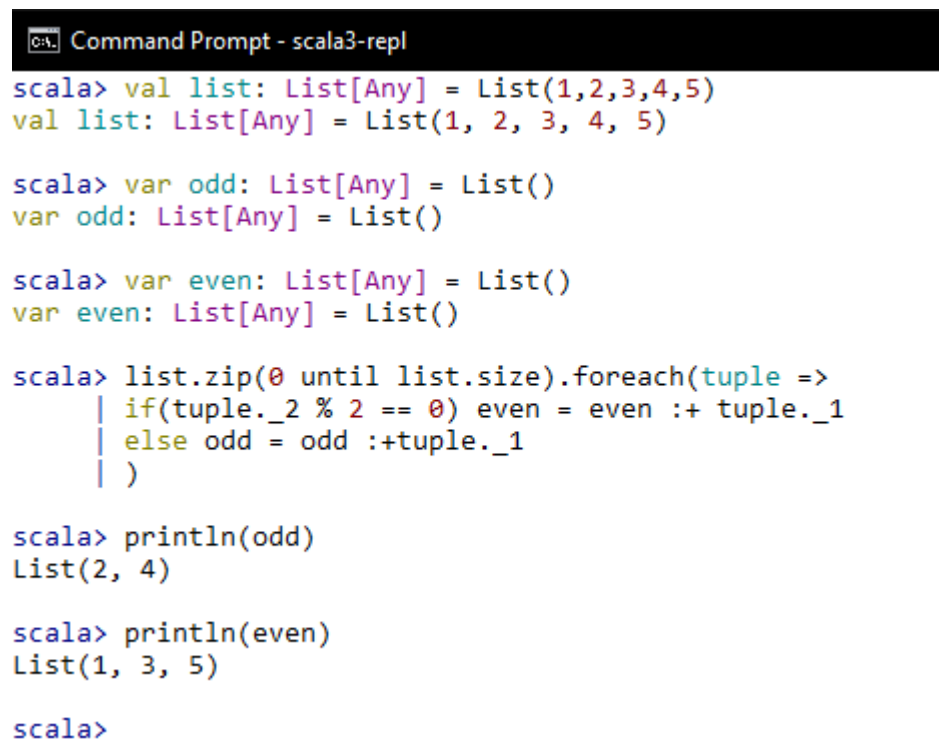
Рисунок 24 – Задание 27

28. $f(n)$ - наибольшее совершенное число не превосходящее n . Совершенным называется натуральное число n равное сумме своих делителей, меньших n , например $6 = 1 + 2 + 3$ ($f(6) = 6$, $f(7) = 6$, ...).

```
scala> def f(n: Int) : Int = {  
  |   return Range(1,n).filter(item => item == Range(1, item).filter(value => item %value == 0).sum).max  
  | }  
def f(n: Int): Int  
scala> f(29)  
val res4: Int = 28
```

Рисунок 25 – Задание 28

29. Список содержит элементы одного, но любого типа. Получить два списка из элементов исходного, выбирая в первый элементы с четными индексами, а во второй с нечетными.



```
scala> val list: List[Any] = List(1,2,3,4,5)  
val list: List[Any] = List(1, 2, 3, 4, 5)  
  
scala> var odd: List[Any] = List()  
var odd: List[Any] = List()  
  
scala> var even: List[Any] = List()  
var even: List[Any] = List()  
  
scala> list.zip(0 until list.size).foreach(tuple =>  
  |   if(tuple._2 % 2 == 0) even = even :+ tuple._1  
  |   else odd = odd :+tuple._1  
  | )  
  
scala> println(odd)  
List(2, 4)  
  
scala> println(even)  
List(1, 3, 5)  
  
scala>
```

Рисунок 26 – Задание 29

30. $f(n)$ - наибольшее из чисел от 1 до n включительно, обладающее свойством: сумма цифр n в некоторой степени > 1 равна самому числу n . Пример: $512 = 8^3$

```
scala> def f( n: Int): Int = {
    |   var count = 1
    |   var sum = 0
    |   for (i <- n to 1 by -1) {
    |     var j = i
    |     while (j>0) {
    |       sum += j%10
    |       j/=10
    |     }
    |     if (sum > 1) {
    |       val k = sum
    |       while (sum < i) {
    |         sum *= k
    |         count +=1
    |       }
    |     }
    |     if (sum == i && count != 1) return sum
    |     sum = 0
    |   }
    |   return 0
    | }
def f(n: Int): Int

scala> f(83)
val res8: Int = 81

scala>
```

Рисунок 27 – Задание 30

31. Список в качестве элементов содержит кортежи типа: (n, s), где n — целые числа, а s — строки. Получить два списка из элементов исходного, выбирая в первый числа, а во второй строки из кортежей.

```
scala> var list: List[Tuple2[Int, String]] = List((1, "first"), (2, "second"))
var list: List[(Int, String)] = List((1,first), (2,second))

scala> var numbers: List[Int] = List()
var numbers: List[Int] = List()

scala> var string: List[String] = List()
var string: List[String] = List()

scala> list.foreach {item =>
    |   numbers = numbers :+ item._1
    |   string = string :+ item._2
    | }

scala> numbers
val res5: List[Int] = List(1, 2)

scala> string
val res6: List[String] = List(first, second)

scala>
```

Рисунок 28 – Задание 31