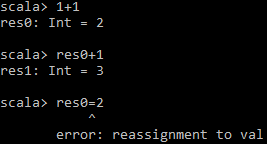
Задания по функциональному программированию

1. Переменные res – это значения val или настоящие переменные var?



2. Язык Scala позволяет умножать строки на числа – попробуйте выполнить выражение "crazy" \* 3 в REPL. Что получилось в результате? Где в Scaladoc можно найти ее описание?



3. Что означает выражение 10 max 2? В каком классе определен метод max?



Класс: math

4. Используя число типа BigInt, вычислите 21024.



179769313486231590772930519078902473361797697894230657273430081157732675805500963132708477322407536021120113879871393357658789768814416622492847430639474124377767893424865485276302219601246094119453082952085005768838150682342462881473913110540827237163350510684586298239947245938479716304835356329624224137216

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

Класс: BigInt.probablePrime, util.Random

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 "qsnvbevtomcj38o06kul". Отыщите в Scaladoc методы, которые можно было бы использовать для этого.

**var** a:BigInt =BigInt.*probablePrime*(100, util.Random)  
*printf*(a.toString(36))

2opc3wrr7hbr0ow0blpv

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

**var** a = "abcd"  
*println*(a.apply(0).toString);  
*println*(a.apply(a.length-1).toString);

a

d

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

take - Selects the first n elements.  
Выбирает первые n элементов.

takeRight - Selects the last n elements.  
Выбирает последние n элементов.

drop - Selects all elements except first n ones.  
Выбирает все элементы, кроме первых n.

drop - Selects all elements except last n ones.  
Выбирает все элементы, кроме последних n.

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

**if** (x>0) 1 **else if** (x<0) -1 **else** 0

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

Unit = ()

11. Напишите на языке Scala цикл, эквивалентный циклу на языке Java for (int i=10; i>=0; i--) System.out.println(i);

**for** (i <- 10 to 0 by -1){  
 *println*(i)  
}

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

**def** countdown (n: Int){  
 **for** (i <- n to 0 by -1){  
 *println*(i)  
 }  
}

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

**var** str ="Hello"  
**var** a: BigInt = 1  
**for** (i <- str){  
 a =i\*a  
}

9415087488

14. Решите предыдущее упражнение без применения цикла.

15. Напишите функцию product(s: String), вычисляющую произведение, как описано в предыдущих упражнениях.

**def** product(s: String)={  
 **var** a: BigInt = 1  
 **for** (i <- s){  
 a =i\*a  
 }  
 a  
}

16. Сделайте функцию из предыдущего упражнения рекурсивной.

**def** product(s: String): BigInt ={  
 **var** a: BigInt=1  
 **if** (s.length>1) {s(0)\**product*(s.drop(1))}**else** s.apply(0)\*a  
}

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

• xn = y2, если n – четное и положительное число, где y=xn/2

• xn = x\*xn-1, если n – нечетное и положительное число.

• x0=1.

• xn=1/x-n, если n – отрицательное число. Не используйте инструкцию return.

**def** powNum(x: BigDecimal, n: Int): BigDecimal ={  
 **if** (n>0) {**if** (n%2==0) x.pow(n/2).pow(2) **else** x\*x.pow(n-1)}**else if** (n==0) 1 **else** 1/x.pow(-n)  
}