Мыслим функционально

## **Задание 1 00:00:01**

Отметьте верные утверждения.

функцию нельзя вызвать дважды

функция должна иметь как минимум один аргумент

функция может иметь побочные эффекты

функция может являться значением переменной

функция имеет тип возвращаемого значения

функция имеет собственные методы

функция может быть передана в качестве параметра в метод

## **Задание 2 00:00:01**

Рассмотрим определение метода map в классе Fraction:

**class** **Fraction**(val numerator: Int,

val denominator: Int) {

**def** map(numF: Int => Int, denF: Int => Int): Fraction =

**new** Fraction(numF(numerator), denF(denominator))

}

Отметьте синтаксически корректные способы вызова метода map (f — это экземпляр класса Fraction).

f.map(\_ + 1, -1)

f.map(n => n + 1, d => d - 1)

f.map(n => 1, d => -1)

f.map(+1, -1)

f.map(\_ + 1, \_ - 1)

f.map({n => n + 1}, {d => d - 1})

## **Задание 3 00:00:01**

Выберите тип функции, соответствующий сокращенной записи (String, String) => Double:

Function[Double, String, String]

Function0[Double, String, String]

Function1[String, String, Double]

Function1[Double, String, String]

Function2[String, String, Double]

Function3[Double, String, String]

Function3[String, String, Double]

Function0[String, String, Double]

Function[String, String, Double]

Function2[Double, String, String]

## **Задание 4 00:00:01**

Функция — это

экземпляр типа FunctionN, где N — количество аргументов

базовый тип JVM

блок из нескольких выражений

альтернативная запись метода с параметрами

## **Задание 5 00:00:02**

Заполните пропуски в приведенных сценариях подходящими функциональными методами.

### **Сценарий 1**

Seq("one", "two", "three", "four", "five")

.\_\_\_\_\_\_(\_.startsWith("t"))

res0: Seq[String] = List(two, three)

### **Сценарий 2**

Seq("one", "two", "three", "four", "five")

.\_\_\_\_\_\_(s => s.substring(0, 1))

res0: Seq[String] = List(o, t, t, f, f)

### **Сценарий 3**

Seq("one two three", "four five")

.{{ flatMap }}(\_.split(" "))

strings.map(\_.length)

res1: Seq[Int] = List(3, 3, 5, 4, 4)