

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Кафедра «Информатика»
кафедра

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2

Классы, объекты, трейты
Тема

Преподаватель

подпись, дата

А. А. Чикизов

инициалы, фамилия

Студент

КИ19-16/16 031939175

номер группы, зачетной книжки

подпись, дата

А. Д. Непомнящий

инициалы, фамилия

Красноярск 2022

1 Задание

Описать трейт для работы с одномерным массивом. Трейт должен включать метод, вычисляющий сумму элементов массива, расположенных между первым и последним нулевыми элементами.

Описать класс, наследующий трейт и реализующий свойство только для чтения для получения количества элементов массива, конструктор заполняющий заданное в его параметре количество элементов массива случайными числами в заданном диапазоне, конструктор, в параметрах которого задаются значения элементов массива (произвольное количество).

Вариант 11.

2 Исходный код основного алгоритма

Исходный код программы приведен на листинге 1.

Листинг 1 — исходный код трейта и класса

```
import scala.util.Random

trait ArrayProcessing:
  def ProcessArray(subject: Array[Int]): Int =
    subject.slice(subject.indexOf(0), subject.lastIndexOf(0)).sum

class ArrayProcessor(val subject: Array[Int]) extends ArrayProcessing :
  def this(n: Int) = this(Array.fill(n) (Random.nextInt(5)-2))

  def this(elements: Int*) = this(elements.toArray)

  private val _count = subject.count(p => true)

  def count: Int = _count

  def printArray(): Unit = subject.foreach(x => print(s"$x, "))

  def ProcessArray(): Int = super.ProcessArray(subject)
```

3 Результаты

Для демонстрации функционала класса была написана программа с кодом, приведенным на листинге 2.

Листинг 2 — исходный код программы для демонстрации

```
package com.github.durakin
package lab2

import scala.sys.exit

@main
def Main(): Unit =
  val c = new ArrayProcessor(2, 0, 4, 7, 0, 7)
  c.printArray()
  println(c.count)
  println(c.ProcessArray())
  val d = new ArrayProcessor(10)
  d.printArray()
  println()
  println(d.ProcessArray())
```

В результате работы программы был получен следующий результат, соответствующий ожидаемому.

```
2, 0, 4, 7, 0, 7, 6
11
0, 0, 1, -1, -1, 0, 1, 1, 2, 2,
-1
```