Лабораторная работа № 1 «Введение в функциональное программирование на языке Scala»

13 марта 2024 г.

Илья Афанасьев, ИУ9-61Б

Цель работы

Целью данной работы является изучение базовых объектно-ориентированных возможностей языка Scala.

Индивидуальный вариант

Мультимножество строк с операциями объединения («+»), пересечения («*») и вычитания («-»). В мультимножестве одна строка может содержаться в нескольких экземплярах.

Реализация и тестирование

```
import scala.collection.immutable.Map

class Multiset(private val _multiples: Map[String, Int]):
  def this (elements: List[String]) =
    this(elements.groupBy(identity).view.mapValues(_.size).toMap)

  def toList: List[String] =
    _multiples
      .map((k, v) => (for i <- 1 to v yield k).toList)
      .reduce(_ ::: _)

  def +(rhs: Multiset) = Multiset(
    _multiples ++
    rhs._multiples
      .map((k, v) => k -> _multiples.getOrElse(k, 0).max(v))
```

```
  )

  def *(rhs: Multiset) = Multiset(
    _multiples
      .map((k, v) => k -> rhs._multiples.getOrElse(k, 0).min(v))
      .filter(_._2 > 0)
  )

  def -(rhs: Multiset) = Multiset(
    _multiples
      .map((k, v) => k -> (v - rhs._multiples.getOrElse(k, 0)))
      .filter(_._2 > 0)
  )

end Multiset

@main def lab2 =
  val m1 = Multiset(List("one", "two", "three", "three", "two", "four", "two"))
  println("m1: " + m1.toList)

  val m2 = Multiset(List("one", "one", "three", "two", "three"))
  println("m2: " + m2.toList)

  println("m1 + m2: " + (m1 + m2).toList)
  println("m1 * m2: " + (m1 * m2).toList)
  println("m1 - m2: " + (m1 - m2).toList)
  println("m2 - m1: " + (m2 - m1).toList)
```

Вывод

В результате выполнения лабораторной работы я изучил базовые объектно-ориентированные возможности языка Scala. Также в процессе выполнения приходилось неоднократно обращаться к документации, в результате чего я познакомился с некоторыми стандартными контейнерными классами и их методами, использовал на практике наиболее употребительные ФВП. Поразительно, как использование здесь лишь некоторых функциональных возможностей языка позволяет абстрактно и лаконично в сравнении с императивным подходом реализовывать идеи.