Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

К У Р С О В А Я Р А Б О Т А

Создание приложения на языке Scala
по дисциплине «Функциональное программирование»

Выполнил
студент гр.3530904/80004 Чугайнов В.А.

Руководитель Лукашин А.А.

«20» декабря 2019 г.

Санкт-Петербург
2019

Оглавление

Задания

1. Калькулятор с приоритетами

Скриншоты

1) Пробельные символы при вводе

```
Enter an expression or press q to quit
12    -    3   +   1
result = 10
```

2) Корректные вводы

```
Enter an expression or press q to quit
12*2- 10*5+4/2
result = -24
```

```
Enter an expression or press q to quit
15 * 0 - 5 + 5 / 5 * 10
result = 5
```

3) Ошибочный ввод

```
Enter an expression or press q to quit
12 + 2o -2
error input
```

4) Конец ввода – символ ‘q’

```
Enter an expression or press q to quit
q
--------------------------------
```

Код программы

```
import scala.io.StdIn.readLine
import scala.annotation.tailrec

object Calculator extends App {

  val reg = """(\d+)(\s*[+-/*]\s*\d+)+""".r

  var cond = true
  while (cond) {
    println("Enter an expression or press q to quit")

    val expr = readLine()
    println(expr match {
      case reg(_, _) =>
        "result = " + calculate(("\\d+".r findAllIn expr).map(_.toInt).toList,
          ("[+-/*]".r findAllIn expr).toList)
      case "q" => {
        cond = false
        "-----------------------------------"
      }
      case _ => "error input"
    })
  }

  def calculate(numList: List[Int], operatorList: List[String]): String = {

    @tailrec def getSumList(sumList: List[Int], numList: List[Int],
operatorList: List[String]): List[Int] = {

      if (operatorList.nonEmpty) {
        operatorList.head match {
          case "-" | "+" => getSumList(numList.head :: sumList, numList.drop(1),
operatorList.drop(1))
          case "*" => getSumList(sumList, (numList.head * numList(1)) ::
numList.drop(2), operatorList.drop(1))
          case "/" => if (numList(1) != 0) {
            getSumList(sumList, (numList.head / numList(1)) :: numList.drop(2),
operatorList.drop(1))
          }
          else {
            throw new Exception("Divide by zero")
          }
        }
      }
      else {
        (numList.head :: sumList).reverse
      }
    }

    val sList = List[Int]()
    val sumList = getSumList(sList, numList, operatorList)
    val isMulti = (op: String) => (op != "/" && op != "*")
    val plusMinusList = operatorList.filter(isMulti)

    def sumExceptFst(l1: List[Int], l2: List[String]): Int = l2 match {
      case el :: rest =>
        if (el == "+") sumExceptFst(l1.drop(1), l2.drop(1)) + l1(1)
        else sumExceptFst(l1.drop(1), l2.drop(1)) - l1(1)
      case Nil => 0
    }

    (sumList.head + sumExceptFst(sumList, plusMinusList)).toString

  }
}
```

Заключение

В данной курсовой работе был использован язык функционального программирования Scala для реализации калькулятора. Работа продемонстрировала преимущества функционального программирования: краткость, лаконичность и быстродействие.