КУРСОВАЯ РАБОТА

по дисциплине «Языки и средства функционального программирования»

**Разработка приложения с использованием принципов функционального программирования**

Выполнил:

Студент гр. 3530904/80003 Королев Д. О

Проверил: Лукашин А. А

**Оглавление**

[Описание задачи 3](#_Toc27609584)

[Описание решения 4](#_Toc27609585)

[Cкриншоты программы 5](#_Toc27609586)

[Вывод 6](#_Toc27609587)

[Исходный код 7](#_Toc27609588)

# Описание задачи

Разработать графическое приложение “Калькулятор” с использованием принципов функционального программирования.

# Описание решения

В ходе алгоритма для получения значения выражения «9 / 3 + 4 \* 6» происходят следующие действия:

1. Формирование из входной строки списка, состоящего из пар вида знак - выражение *(case class Token(nextOp: String, value: Expression*), на вход *tokenize* передается *input.split(" ").toList*

|  |
| --- |
| object Tokenizer {  def tokenize(input: List[String]): Unparsed = input match {  case Nil => Nil  case n :: rest => rest match {  case Nil => List(Token("", Number(n.toInt)))  case "+" :: tail => new Token("+", Number(n.toInt)) :: tokenize(tail)  case "-" :: tail => new Token("-", Number(n.toInt)) :: tokenize(tail)  case "\*" :: tail => new Token("\*", Number(n.toInt)) :: tokenize(tail)  case "/" :: tail => new Token("/", Number(n.toInt)) :: tokenize(tail)  case \_ => List(Token("", Number(n.toInt)))  }  } } |

1. Парсинг и приведение выражения «9 / 3 + 4 \* 6» к виду

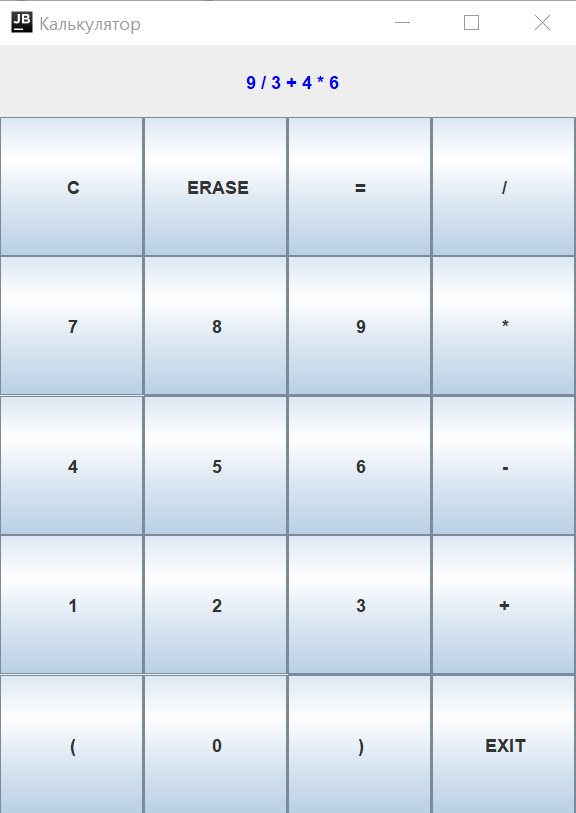
*Add(Divide(Number(9),Number(3)),Multiply(Number(4),Number(6)))*

|  |
| --- |
| def getAgg(op: String): (Expression, Expression) => Expression = op match {   case "+" => (x: Expression, y: Expression) => Add(x,y)  case "-" => (x: Expression, y: Expression) => Subtract(x,y)  case "\*" => (x: Expression, y: Expression) => Multiply(x,y)  case "/" => (x: Expression, y: Expression) => Divide(x,y) } |

1. Вычисление конечного значения выражения

|  |
| --- |
| def calculate: Int = this match {  case Number(n) => n  case Multiply(l, r) => l.calculate \* r.calculate  case Divide(l, r) => l.calculate / r.calculate  case Subtract(l, r) => l.calculate - r.calculate  case Add(l, r) => l.calculate + r.calculate } |

# Cкриншоты программы



­­­

# Вывод

В ходе выполнения курсовой работы было реализовано графическое приложение «Калькулятор» и закреплены навыки применения принципов функционального программирования, которые являются крайне удобными и эффективными при написании программ.

# Исходный код

https://github.com/mycelium/hsse-fp-2019-2/tree/3530904/80003\_denis-korolev/tasks/courceProject