ОТЧЕТ ПО КУРСОВОЙ РАБОТЕ

КУРСОВАЯ РАБОТА

по дисциплине «Языки и средства функционального программирования»

**Разработка приложения с использованием принципов функционального программирования**

Выполнил:

Студент гр. 3530904/80003 Фирсов Д.А.

Руководитель: Лукашин А.А.

Оглавление

[1. Техническое задание 3](#_Toc27602319)

[2. Цели и Задачи 3](#_Toc27602320)

[2.1. Цель 3](#_Toc27602321)

[2.2. Задачи 3](#_Toc27602322)

[3. Выполнению работы 3](#_Toc27602323)

[3.1 Подготовка 3](#_Toc27602324)

[3.2 Выполнение 4](#_Toc27602325)

[4. Пример работы 4](#_Toc27602326)

[5. Вывод 5](#_Toc27602327)

[6. Приложение 5](#_Toc27602328)

1.  Техническое задание

|  |
| --- |
| Курсовик (расчетно-графическое задание)Простой  * Данные варианты курсовика рекомендованы для тех, кому совсем неинтересно и хочется потратить минимум времени для его написания:   1. Реализовать интерфейс к лабораторной Scala-3      + Ввод предложения      + Вывод анаграммы      + Редактирование словаря   2. Аналогично 1, но вместо интерфейса использовать файлы   3. Калькулятор |

1. Мной был выбран вариант “Калькулятор”

2. Цели и Задачи

2.1. Цель

1. Калькулятор

2.2. Задачи

1. Реализация калькулятора в функциональном стиле на языке Scala;

2. Применение графики к реализованному функционалу;

3. Выполнению работы

3.1 Подготовка

1. Скачал и установил IntelliJ Idea 2019
2. Скачал и установил OpenJDK13
3. В IntelliJ Idea установил плагин Scala
4. Для графики использовал библиотеку Swing, пришлось добавить в build.sbt необходимые зависимости.

3.2 Выполнение

1. Написал калькулятор с приоритетами
2. Для реализации приоритетов пришлось преобразовать инфиксную запись выражения в обратную польскую нотацию с использованием стека.
3. На вход подается строка, из которой с помощью *inputSanityCheck(userInput)* проверяется на корректность ввод пользователя, и преобразуется в необходимый параметр для *convertFromInfixToReversePolish(inputLst)*
4. evaluateReversePolishInput(inputList: List[String]) вычисляет выражение в обратной польской нотации и возвращает его результат.

4. Пример работы

1. Пользователь с помощью графического интерфейса вводит выражение, которое требуется вычислить.
2. По нажатию на кнопку « = » запускается анализ выражения. Преобразуется в строку. Из нее парсится выражение для обработки. Если пользователь ввёл некорректное выражение, то выводится строка “Error”.
3. В случае корректного ввода результат вычисляется и выводится на экран.

|  |  |
| --- | --- |
|  |  |
|  |  |

5. Вывод

Я написал графическое приложение «калькулятор», используя подходы функционального программирования познакомился с библиотекой Swing в Scala. Также изучил некоторые алгоритмы функционального программирования.

6. Приложение

Ссылка на Git [hhttps://github.com/mycelium/hsse-fp-2019-2/tree/3530904/80003\_daniil-firsov/tasks/scala/CourseProject/CalcGUI](https://github.com/mycelium/hsse-fp-2019-2/tree/3530904/80003_daniil-firsov/tasks/scala/CourseProject/CalcGUI)