КУРСОВАЯ РАБОТА

по дисциплине «Языки и средства функционального программирования»

**Разработка приложения с использованием принципов функционального программирования**

Выполнила:

Студент гр. 3530904/80003 Беседин Д.Д.

Проверил: Лукашин А. А.

**Оглавление**

[Введение 3](#_Toc27925838)

[Задача 4](#_Toc27925839)

[Исходный код 5](#_Toc27925840)

[Вывод 5](#_Toc27925841)

# Введение

Функциональное программирование – это ветвь программирования, при котором программирование ведется с помощью определения функций. Оно основывается на нескольких важных концепциях: отсутствие побочных эффектов и изменяемых данных, чистые функции и их композиция.

Важным преимуществом реализации языков функционального программирования является автоматизированное динамическое распределение памяти компьютера для хранения данных. При этом программист избавляется от рутинной необходимости контролировать данные, а при необходимости может запустить функцию «сборки мусора» - очистки памяти от тех данных, которые больше не потребуются программе.

# Задача

Разработать приложение с использованием принципов функционального программирования.

# Исходный код

import scala.util.{Try, Success, Failure}

def calculator(left: String, op: String, right: String): Unit = {

def parse(value: String) = Try(value.toDouble)

(parse(left), parse(right)) match {

case (Success(leftDouble), Success(rightDouble)) =>

op match {

case "/" => println(leftDouble / rightDouble)

case "\*" => println(leftDouble \* rightDouble)

case "+" => println(leftDouble + rightDouble)

case "-" => println(leftDouble - rightDouble)

case invalid: String => println(s"Invalid operator $invalid.")

}

case (Failure(\_), \_) => println(s"Invalid operand $left.")

case (\_, Failure(\_)) => println(s"Invalid operand $right.")

case (Failure(\_), Failure(\_)) => println(s"Invalid operands $left and $right.")

}

}

# Вывод

В ходе выполнения курсовой работы были закреплены навыки использования принципов функционального программирования.