Министерство образования и науки РФ

Санкт-Петербургский Политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа программной инженерии

**КУРСОВАЯ РАБОТА**

**ПО ДИСЦИПЛИНЕ «ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ»**

КАЛЬКУЛЯТОР НА ЯЗЫКЕ ФУНКЦИОНАЛЬНОГО ПРОГРАММИРОВАНИЯ HASKELL

Выполнил:

Студент гр. 3530904/80001 Трофимов Ф.О.

Руководитель: Лукашин А.А.

**Оглавление**

[Введение 3](#_Toc27889411)

[Задание 3](#_Toc27889412)

[Решение 4](#_Toc27889413)

[Проверка работы 4](#_Toc27889414)

[Вывод 5](#_Toc27889415)

# **Введение**

Функциональное программирование — это практическая реализация идей Алонзо Чёрча. Не все идеи лямбда-исчисления переросли в практическую сферу, так как лямбда-исчисления не учитывали физических ограничений. Тем не менее, как и ОО программирование, функциональное программирование — это набор идей, а не набор четких указаний. Существует много функциональных языков, и большинство из них делают одни схожие вещи по-разному.

Лямбда исчисление было придумано для изучения проблем, связанных с вычислениями. Функциональное программирование, стало быть, в первую очередь имеет дело с вычислениями, и, на удивление, использует для этого функции. *Функция* — это базовый элемент функционального программирования. Функции используются почти для всего, даже для простейших расчётов. Даже переменные заменяются функциями. В функциональном программировании переменные — это просто синонимы для выражений (чтобы нам не пришлось писать всё в одну строку). Их нельзя изменять. В каждую переменную можно записать только один раз.

# **Задание**

Реализовать калькулятор арифметических выражений на языке *Haskell*.

# **Решение**

С помощью «парсинга» определяется конкретное значение введенного выражения, а с помощью *evalExpr* – конкретная операция, после чего производится непосредственно вычисление значения выражения. Например, приведем часть исходного кода:

|  |
| --- |
| op\_term :: Int -> Expr -> Parser Expr  op\_term n t1 =  do symbol "+"  t2 <- term n  o <- op\_term n (Add t1 t2)  return o  +++ do symbol "-"  t2 <- term n  o <- op\_term n (Sub t1 t2)  return o  +++ return t1 |

# **Проверка работы**

****

Рисунок . Выполнение арифметических операций

# **Вывод**

В результате выполнения работы был реализован калькулятор арифметических выражений на языке *Haskell* с использованием парадигмы функционального программирования. Калькулятор поддерживает операции сложения, вычитания, умножения, деления и другие.

Использование языка *Haskell* для выполнения работы позволило познакомиться с разработкой приложений в функциональном стиле и расширить свои знания в этой сфере программирования.