

Санкт-Петербургский Политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа программной инженерии

КУРСОВАЯ РАБОТА

по дисциплине «Функциональное программирование»

Калькулятор на Haskell

Выполнил:

Студент гр. 3530904/80001

Шахбазлы Т.Э.о

Руководитель:

Лукашин А.А.

Санкт-Петербург

2019 г.

О г л а в л е н и е

Введение	3
Задание	3
Решение	3
Проверка работы	4
Вывод	4

В в е д е н и е

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 (*alias*) для выражений (чтобы нам не пришлось писать всё в одну строку). Их нельзя изменять. В каждую переменную можно записать только один раз.

З а д а н и е

Реализовать калькулятор арифметических выражений на языке Haskell.

Р е ш е н и е

С помощью парсинга определяется конкретное значение введенного выражения, а с помощью `evalExpr` — конкретная операция, после чего производится непосредственно вычисление значения выражения. К примеру,

```
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
```

*весь исходный код представлен здесь:

https://github.com/mycelium/hsse-fp-2019-2/tree/3530904/80001_shahbazly.tural/tasks

Проверка работы

Выполнение арифметических операций:

```
ghc -o main Parsing.hs main.hs
./main
42
42
42 = 42
= 42
42 =

Repl.it: Stopping... please wait
exited, terminated
> 1
=> 1
> 1 + 1
=> 2
> 1 * 2
=> 2
> 1 + 1 -
=> 2
> 10 * 10
=> 100
> 10 / 10
=> 1.0
> 10 / 100
=> 0.1
> 10000
=> 10000
> 10000*1341134
=> 13411340000
```

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.