

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
Санкт-Петербургский политехнический университет Петра Великого  
Кафедра «Компьютерных наук и технологий»  
Высшая школа программной инженерии

Курсовой проект  
по дисциплине: Языки и средства функционального программирования  
название: Калькулятор на Haskell

Выполнил студент  
группы 3530904/80007  
Дроздов Кирилл Константинович  
Проверил преподаватель:  
Лукашин Алексей Андреевич

Санкт-Петербург  
2019

## Оглавление

<u>Введение.....</u>	<u>3</u>
<u>Задание.....</u>	<u>4</u>
<u>Решение.....</u>	<u>4</u>
<u>Скриншот.....</u>	<u>4</u>
<u>Вывод.....</u>	<u>5</u>

## Введение

Функциональное программирование — это стиль программирования, который опирается на вычисление выражений, а не на выполнение команд. Выражения формируются посредством комбинирования функций. Функциональный язык — это язык, который поддерживает программирование в функциональном стиле.

Функциональные языки бывают чистыми и нечистыми. Чистые функциональные языки сравнительно редки, к ним относятся в первую очередь Haskell и Clean. В чистых языках нет побочных эффектов. На практике это означает отсутствие присваивания и ввода-вывода в том виде, к которому мы привыкли. Это создает ряд трудностей, хотя в уже упомянутых языках это решено довольно хитроумно, и на этих языках пишут код с большим количеством ввода-вывода. Языки типа Lisp, OCaml или Scala допускают функции с побочными эффектами, и в этом смысле данные языки зачастую более практичны.

## Задание

Реализовать калькулятор арифметических выражений на языке Haskell.

## Решение

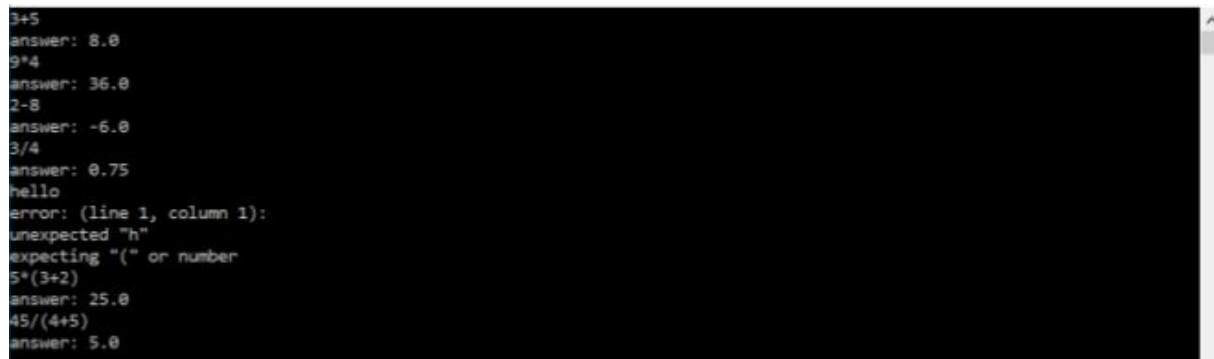
Parsec обеспечивает разбиение выражения. Выражения определяются `buildExpressionParser`. Функция `flip` изменяет порядок аргументов в передаваемой функции. Далее идет разбиение на унарный или бинарный операторы (префиксный или инфиксный).

```
parseExpression :: Parser Double
parseExpression = (flip buildExpressionParser) parseItem $ [
    [ Prefix (reservedOp lexer "-" >> return negate) ],
    [ Infix  (reservedOp lexer "+" >> return (+)) AssocLeft,
      Infix  (reservedOp lexer "-" >> return (-)) AssocLeft],
    [ Infix  (reservedOp lexer "*" >> return (*)) AssocLeft,
      Infix  (reservedOp lexer "/" >> return (/)) AssocLeft]
] --evaluate the function flipping the order of arg
```

Исходный код:

[https://github.com/mycelium/hsse-fp-2019-2/blob/3530904/80007\\_Drozдов-Kirill/calculator.hs](https://github.com/mycelium/hsse-fp-2019-2/blob/3530904/80007_Drozдов-Kirill/calculator.hs)

## Скриншот



```
3+5
answer: 8.0
9*4
answer: 36.0
2-8
answer: -6.0
3/4
answer: 0.75
hello
error: (line 1, column 1):
unexpected "h"
expecting "(" or number
5*(3+2)
answer: 25.0
45/(4+5)
answer: 5.0
```

## **Вывод**

В результате данной курсовой работы был создан проект калькулятора на функциональном языке программирования Haskell. Данные поступают через стандартный поток ввода. Результат выводится в стандартный поток вывода.

Калькулятор поддерживает стандартные операции сложения, вычитания, умножения и деления. Также одноуровневые скобочные конструкции. При вводе символов, которые не поддерживаются калькулятором, выводится сообщение об ошибке.