ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский политехнический университет Петра Великого»

Институт компьютерных наук и технологий

Курсовой проект по дисциплине

«Функциональное программирование»

Выполнил студент

гр. 3530904/80005: Власенко Н.А.

Руководитель

ассистент ВШПИ: Лукашин А. А.

Санкт-Петербург

2019

Оглавление

[1. Постановка задачи 3](#_Toc27690341)

[2. Ход работы 3](#_Toc27690342)

[2.1. Текст программы 3](#_Toc27690343)

[2.2 Пример выполнения 4](#_Toc27690344)

[2.3. Репозиторий 4](#_Toc27690345)

[3. Вывод 4](#_Toc27690346)

# Постановка задачи

Калькулятор, поддерживающий простые арифметические операции: функции сложения, вычитания, умножения и деления, остатка от деления при использовании языка программирования Haskell.

# Ход работы

## 2.1. Текст программы

import Control.Applicative ((<$>), (<\*>))

type Operator = Double -> Double -> Double

type Entry = (String, Operator)

type Register = [Entry]

modulu :: Double -> Double -> Double

modulu a b = fromIntegral $ mod (round a) (round b)

operatorRegister :: Register

operatorRegister = [

("-", (-)),

("+", (+)),

("/", (/)),

("\*", (\*)),

("%", modulu)

]

main = print $ calculate "55 \* 2 + 5 % 2"

calculate :: String -> Maybe Double

calculate = eval operatorRegister . words

eval :: Register -> [String] -> Maybe Double

eval [] \_ = Nothing -- No operator found.

eval \_ [] = Nothing -- If a operator don't have anything to operate on.

eval \_ [number] = Just $ read number

eval ((operator, function):rest) unparsed =

case span (/=operator) unparsed of

(\_, []) -> eval rest unparsed

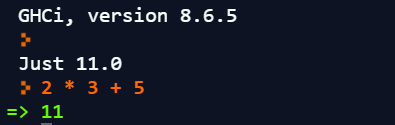
(beforeOperator, afterOperator) ->

function

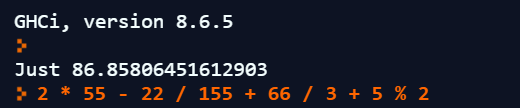
<$> (eval operatorRegister beforeOperator)

<\*> (eval operatorRegister $ drop 1 afterOperator)

## 2.2 Пример выполнения







## 2.3. Репозиторий

<https://github.com/mycelium/hsse-fp-2019-2/tree/3530904/80005_vlasenko-natalia/coursework>

# Вывод

В ходе работы был изучен функциональный подход к программированию, изучены некоторые основные алгоритмы языка Haskell.