

Курс: Функциональное программирование
Практика 4. Введение в Haskell.

Разминка

► Запустите интерпретатор GHCi и вычислите значения следующих выражений:

```
5 * (-3)
5 * -3
5*-3
```

```
5 == 7
's' /= 'S'
5 == True
```

```
succ 41
succ 'z'
succ pi
```

```
div 42 10
42 `div` 10
42 `rem` 10
```

```
10001000
7(-1)
7**(-1)
```

```
sqrt 2
sqrt (-2)
```

```
sqrt (sqrt 16)
sqrt $ sqrt 16
```

```
undefined
error "AAA!!!!111"
```

► В GHCi определите тип следующих выражений:

```
7
7.0
```

```
sqrt 25
sqrt 25.0
```

```
(+)
(40 +)
(+ 2)
40 + 2
```

```
succ
```

```
succ 'z'  
succ 41  
succ pi  
  
undefined  
error  
error "AAA!!!!111"
```

► Определите тип функций:

```
f1 x = (+) x 3  
f2 = \x -> (+) x 3
```

Определение функций

- Реализуйте функцию, утраивающую значение своего аргумента.
- Реализуйте функцию, возвращающую знак числа (1, 0 или -1).
- Реализуйте следующие двухместные логические операции: стрелку Пирса (Peirce's arrow, NOR) и штрих Шеффера (Sheffer stroke, NAND).
- Реализуйте функцию, находящую числа Фибоначчи

$$a_0 = 0; a_1 = 1; a_{k+2} = a_k + a_{k+1}.$$

Какова её сложность?

Стандартные функции

- Стандартная функция `flip` принимает функцию двух аргументов и возвращает эту же функцию, но с переставленными местами аргументами. Попробуйте записать (1) тип, (2) код этой функции. Найдите реализацию этой функции в стандартной библиотеке и сравните со своей версией.
- Знаете ли вы функцию с типом `s -> t -> s`? Найдите такую функцию с помощью Google. Просмотрите её определение.
- Определите с помощью Google в каком модуле находится функция `fix` (комбинатор неподвижной точки), просмотрите её определение.
- Используя комбинатор неподвижной точки `fix`, напишите «нерекурсивное» определение факториала.
- Запишите комбинатор «сумма квадратов двух величин» в терминах комбинатора `op`.
- Каково значение выражений (попробуйте ответить без помощи интерпретатора)

```

until (>100) (*3) 1
until (>100) (4*) 1
until odd ('div' 2) 800

```

Факториал может быть выражен через `until` и пары

```

fac n = fst $ until (\x -> snd x >= n) ss zz
  where zz = (1,0)
        ss p = (f*(s+1),s+1)
              where f = fst p
                    s = snd p

```

Попробуйте реализовать подобным образом эффективный алгоритм вычисления чисел Фибоначчи (совет: используйте трёхэлементные кортежи).

Домашнее задание

► (1 балл) Реализуйте функцию, описывающую плотность равномерного распределения на промежутке от 3 до 4.

► (1 балл) Реализуйте функцию, находящую наибольший общий делитель двух целых чисел с помощью алгоритма Евклида:

$$\begin{aligned} \text{GCD}(a, b) &= \text{GCD}(b, a \bmod b), \text{ если } b \neq 0; \\ \text{GCD}(a, 0) &= a. \end{aligned}$$

► (2 балла) Реализуйте функции, находящие сумму и количество цифр заданного целого числа.

► (2 балла) Реализуйте функцию, находящую элементы следующей рекуррентной последовательности

$$a_0 = 1; a_1 = 2; a_2 = 3; a_{k+3} = a_{k+2} + a_{k+1} - 2a_k.$$

Попытайтесь найти эффективное решение.

► (2 балла) Реализуйте функцию, находящую значение определённого интеграла от заданной функции на заданном интервале по методу трапеций.

► (2 балла) Реализуйте функцию, находящую значение квадратного корня методом Ньютона.