

## Практическое занятие «Хаскелл-2»

10 ноября 2020 года

При решении задач данного задания, попытайтесь уменьшить количество используемых операторов `if`. Используйте варианты функции с сопоставлением по шаблону, гарды.

1. Напишите функцию `dFact :: Integer -> Integer`, вычисляющую по неотрицательному целому числу его двойной факториал:

$$n!! = \begin{cases} 1 \cdot 3 \cdot 5 \cdot \dots \cdot n, & n \text{ — нечётное число,} \\ 2 \cdot 4 \cdot 6 \cdot \dots \cdot n, & n \text{ — чётное число.} \end{cases}$$

2. Напишите функцию `sumOfDigits :: Integer -> Int`, которая по целому числу вычисляет сумму цифр его десятичной записи.
3. Напишите функцию `powOf2 :: Integer -> Int`, которая проверяет, является ли заданное положительное целое число степенью двойки и возвращает эту степень. Если число не является степенью двойки, функция должна возвращать `-1`.
4. За каждый пункт этой задачи полагается один балл. В каждом пункте следует написать две функции, вычисляющую указанное выражение: обычную и в безаргументном стиле. Величины  $x$  и  $y$  — целые положительные числа. При написании безаргументного варианта используйте не  $\lambda$ -функции, а конструкции из замыканий функций и функции «flip», «.», «\$». Возможно использование и других средств из библиотеки Хаскелла, но их надо будет объяснить преподавателю.

а)  $f(x, y) = x^y$ ;

б)  $f(x, y) = (x + 1)^y$ ;

в)  $f(x, y) = x^{y-3}$ ;

г)  $f(x, y) = (x + 1)^{y-3}$ .

5. Напишите функцию, принимающей список чисел и вычисляющей сумму их квадратов. По необходимости используйте лямбда-функции, функции «\$» и «.», замыкания функций. Сигнатура функций имеет вид `Num a => [a] -> a`.
  - а) Создайте вариант `sumSquares1` с рекурсией.
  - б) Создайте вариант `sumSquares2` с итерацией (хвостовой рекурсией).
  - в) Создайте вариант `sumSquares3` с функциями высших порядков.
6. Назовём максимумом из нескольких строк, имеющих одинаковую длину и составленных из заглавных латинских букв, новую строку той же длины, такую, что в каждой позиции стоит символ, совпадающий с наибольшим (в алфавитном порядке) из символов в этой позиции исходных строк. Напишите функцию `posMax :: [String] -> String`, которая по списку строк одинаковой длины, вычисляет их максимум.

7. Задан список попарно различных точек, представленных парами своих координат. Гарантируется, что среди этих точек нет начала координат. Напишите функцию

```
minAngle :: [(Double,Double)] -> (Double,Double),
```

которая по списку точек, представленных двухэлементными списками своих координат, выдает точку, имеющую наименьший полярный угол из диапазона  $[0, 2\pi)$ . При равенстве полярных углов двух точек, выдайте ту, которая находится дальше от начала координат. (При вычислениях применяйте функцию `atan2 y x`, выдающую полярный угол точки с координатами  $(x;y)$  из диапазона  $[-\pi; \pi)$ .) Разумно использовать функцию `maximumBy`, находящую максимум среди элементов списка с использованием собственного порядка, задаваемого *компаратором*, функцией, сравнивающей два элемента и выдающей результат в виде значения типа `Ordering`: `LT` — если первый аргумент меньше второго, `EQ` — если аргументы равны, и `GT` — если первый аргумент больше второго. Например, компаратор сравнимых объектов (принадлежащих множеству типов `Ord`) для сортировки по убыванию может иметь вид

```
decComp :: Ord a => a -> a -> Ordering
decComp a b
  | a > b      = LT
  | a == b     = EQ
  | otherwise = GT
```