

Практическое занятие «Хаскелл–7. Функторы, аппликативные функторы» 03 декабря 2019 года

При решении задач практики запрещается использовать *рукописную* рекурсию и функции высших порядков (если не оговорено противное).

1. Конструктор пары $(,)$ — тоже функция с типом $(,) :: a \rightarrow b \rightarrow (a,b)$. По аналогии с задачей про алгебраическую сумму двух списков напишите функцию построения декартова произведения двух списков.
2. Поскольку `Maybe` — это функтор, то преобразование `Just 2` \rightarrow `Just 5` можно осуществить с помощью функции `(+3) <$> Just 2`. При этом, по свойствам `Maybe` имеем, что `(+3) <$> Nothing` \rightarrow `Nothing`. А как увеличить на 3 каждый элемент списка типа `[Maybe Int]`?

Подсказка: поскольку «чистую» функцию увеличения на 3 надо протянуть сквозь два функтора — списка и `Maybe`, то надо обдумать композицию двух функций `<$>`.

3. Пара — это тоже функтор! При этом параметризующий тип — это тип второго элемента пары. Пусть пара типа `(String,Int)` задаёт школьника (первый элемент — имя, второй — рост). Напишите функцию, которая принимает описание школьника и величину, на сколько он подрос за лето, и возвращает описание школьника с новым ростом. Напишите ту же функцию для случая, когда рост — это первый элемент пары. (Рассмотрите функцию `swap :: (a,b) -> (b,a)`, определённую в библиотеке `Data.Tuple`.)
4. Используя опыт двух предыдущих задач, напишите функцию, которая увеличивает на заданную величину рост всех школьников из класса, заданного списком пар, представляющих человека.
5. Пусть точка (x,y) на плоскости задается типом данных

```
data TPoint = Point { x :: Double, y :: Double },
```

а прямая l с уравнением $Ax + By + C = 0$ задается алгебраическим типом данных

```
data TLine = Line { a :: Double, b :: Double, c :: Double }.
```

Напишите функцию, которая принимает список прямых и находит максимум (в лексикографическом порядке) среди точек пересечения прямых из этого набора. Считаем, что параллельные и накладывающиеся прямые не пересекаются. Как следствие, типом результата функции пересечения двух прямых разумно сделать `Maybe TPoint` (поскольку никакой точки в результате может не получиться). Этот же тип разумно положить как тип результата создаваемой функции, поскольку может получиться так, что никакие две прямые из набора не пересекаются и нет вообще ни одной точки пересечения.

Конечно, в рамках этой задачи надо применять функции работы со списками, но постарайтесь максимально использовать возможности, предоставляемые функторами, аппликативными функторами и монадами в Хаскелле (список, пара, `Maybe`). По крайней мере, воздержитесь от рукописной рекурсии :)