

PODSTAWY PROGRAMOWANIA DEKLARATYWNEGO

HASKELL

Ćwiczenia 7

Zadania powtórkowe (20p.)

Zadanie 1.

Podać typ danej funkcji. Obowiązuje znajomość wszystkich funkcji omówionych na wykładzie.
Np. typ funkcji map:

`map :: (a -> b) -> [a] -> [b]`

Podać wartości danej funkcji. Np.

Wartością `map (>1) [3,2,1,0,-1]` jest `[True,True,False,False,False]`

Wartością `map (*2) [3,2,1,0,-1]` jest `[6,4,2,0,-2]`

Zadanie 2.

Napisać definicję funkcji obliczającej n -ty wyraz ciągu danego wzorem $a_n = a_{n-1} + 2a_{n-2}$ ($a_1 = 0$, $a_2 = 5$) a) rekurencyjną, b) w wersji „akumulatorowej”.

Zadanie 3.

Napisać definicję funkcji, która w liście przestawia a) pierwszy element z drugim, b) pierwszy element z ostatnim, c) drugi element z przedostatnim.

Zadanie 4.

Napisać definicję funkcji, której wartością jest liczba wystąpień elementu d w liście l .

Zadanie 5.

Napisać funkcję sprawdzającą równość dwóch list.

Zadanie 6.

Napisać funkcję sprawdzającą równość dwóch zbiorów.

Zadanie 7. (8p.)

Napisać funkcję, która porządkuje niemalejąco elementy listy liczbowej wykorzystując algorytm sortowania a) przez proste wstawianie, b) „bąbelkowego”.

Zadanie 8.

Napisać funkcję, która dla dwóch uporządkowanych niemalejąco list liczbowych *l1* i *l2* daje w wyniku uporządkowaną niemalejąco listę elementów z list *l1* i *l2*.

Zadanie 9.

Napisać funkcję, która dla dwóch drzew binarnych *d1* i *d2* zwraca wartość *True*, gdy drzewo *d1* jest poddrzewem drzewa *d2*.

Zadanie 10. (3p.)

Napisać funkcję, której wartością jest długość a) najdłuższej, b) najkrótszej gałęzi w drzewie binarnym.

Zadanie 11.

Poniższa funkcja ma sprawdzać, czy podana lista jest pusta. Dlaczego definicja ta jest niepoprawna? Jak należy poprawić?

```
pustal :: [a] -> Bool
pustal x = (x == [])
```

Zadanie 12.

Poniższy program ma konwertować ciągi cyfr (String) na liczby (Int). Wyjaśnić, na czym polega błąd w definicji i ten błąd usunąć.

```
import Data.Char

naInt :: String -> Int
naInt = foldl (\x acc -> acc * 10 + digitToInt x) 0
```

Zadanie 13.

Dla danej listy *l* wypisać listę par [(element, liczba wystąpień), ...].

Przykład: w [2,3,4,1,2,5,3,2,4,4,2] = [(2,4),(3,2),(4,3),(1,1),(5,1)]

w ['a','a','b','a'] = [('a',3),('b',1)]

Zadanie 14. (4p.)

Dla danej listy par (x,y) należy zwrócić listę tych par posortowaną wg odległości od punktu (0, 0).

Przykład: f [(0,3),(1,1),(2,1),(1,0)] = [(1,0),(1,1),(2,1),(0,3)]

Zadanie 15. (5p.)

Wartością funkcji *prime* jest lista liczb pierwszych. Napisać:

- a) funkcję, której wartością będzie lista par, w których drugi element będzie liczbą pierwszą, a pierwszy jej numerem, np. `pierwsze = [(1,2),(2,3),(3,5),(4,7),...]`
- b) funkcję znajdującą liczbę pierwszą o podanym numerze, np. `p 3 = 5`.

```
primes = filterPrime [2..]
```

```
where filterPrime (p:xs) =
```

```
  p : filterPrime [x | x <- xs, x `mod` p /= 0]
```

Uwaga:

Wykonane zadania **7, 10, 14, 15** należy przekazać do **19.04.2018, 13:30** przez OLAT „Ćwiczenia 7”. Wszystkie definicje funkcji mają być zapisane w jednym pliku .hs z numerem zadania w komentarzu. Nazwa pliku ma zawierać nazwisko Studenta i numer ćwiczeń.