

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інтелектуальних технологій

Лабораторна робота №3
З дисципліни “Непроцедурне програмування”
Варіант №7

Виконав студент групи КН-31
Пагарський О. А.

Перевірів:
Миколайчук Р. А.

Київ – 2021

Мета: Набути досвіду визначення та використання функцій вищого порядку.

Завдання 1. Визначити частоту кожного елемента списку, напр.: "aaabbcaadddd" ⇒ [('a',5), ('b',2), ('c',1), ('d',4)].

```
import Data.Map as Map
import Data.List as List

-- З застосуванням вбудованих функцій вищого порядку:
frequency :: (Ord ch, Num k) => [ch] -> [(ch, k)]
frequency str = Map.toList $ Map.fromListWith (+) [(c, 1) | c <- str]

-- Без застосування вбудованих функцій вищого порядку:
map' :: Ord k => (a -> k) -> [a] -> [k]
map' _ [] = []
map' f (x:xs) = f x : map' f xs

frequency2 str = map' (\str -> (head str, length str)) $ group $ sort str
```

Тестування:

```
-- Prelude> :load lab3/lab3.hs
-- [1 of 1] Compiling Lab3                ( lab3/lab3.hs, interpreted )
-- Ok, one module loaded.
-- *Lab3> frequency "aaabbcaadddd"
-- [( 'a',5), ( 'b',2), ( 'c',1), ( 'd',4)]
-- *Lab3> frequency "Hello, World!"
--
-- [( ' ',1), ( '!',1), ( ',',1), ( 'H',1), ( 'W',1), ( 'd',1), ( 'e',1), ( 'l',3), ( 'o',2), ( 'r',
1)]
-- *Lab3> frequency ""
-- []
```

Завдання 2. Знайти прості дільники числа.

```
-- З застосуванням вбудованих функцій вищого порядку:
primeFactors :: Int -> [Int]
primeFactors 1 = []
primeFactors n
    | factors == [] = [n]
    | otherwise     = factors ++ primeFactors (n `div` (head factors))
    where factors   = List.take 1 $ List.filter (\x -> mod n x == 0) [2..n-
1]

-- Без застосування вбудованих функцій вищого порядку:
primes = f [2..]
    where f (p:xs) = p : f [x | x <- xs, mod x p /= 0]

primeFactors2 x = f x (head primes) (tail primes)
    where f x n ns
        | x < 2      = []
        | x < n ^ 2  = [x]
```

```
| mod x n == 0 = n : f (div x n) n ns
| otherwise   = f x (head ns) (tail ns)
```

Тестування:

```
-- Prelude> :load lab3/lab3.hs
-- [1 of 1] Compiling Lab3           ( lab3/lab3.hs, interpreted )
-- Ok, one module loaded.
-- *Lab3> primeFactors 15
-- [3,5]
-- *Lab3> primeFactors 27
-- [3,3,3]
-- *Lab3> primeFactors 60
-- [2,2,3,5]
-- *Lab3> primeFactors 2
-- [2]
```

Висновок:

В результаті виконання лабораторної роботи було створено два варіанти для двох функцій (з використанням вбудованих функцій вищого порядку, та без) відповідно з заданим варіантом роботи. Це дозволило закріпити знання та набути практичних навичок по використанню функцій вищого порядку функціональної мови програмування haskell.