**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра інтелектуальних технологій**

**Лабораторна робота №1**

з дисципліни «Непроцедурне програмування»

Тема роботи: «Мова Haskell. Робота з iнтерпретатором ghci»

**Варіант 4**

Виконав студент групи КН-31

Кучерук Владислав

**Київ – 2021**

**Мета роботи**

Ознайомитись з основними типами мови. Ознайомитись зi структурою та

функцiями Glasgow Haskell Compiller. Набути навичок роботи з iнтерпретатором

ghci та визначення найпростiших функцiй.

**Завдання**

1. Наведiть приклади виразiв вказаного типу. Кожен список має мiстити кiлька елементiв. Перегляньте тип прикладiв, як їх визначає ghci. Прокоментуйте.

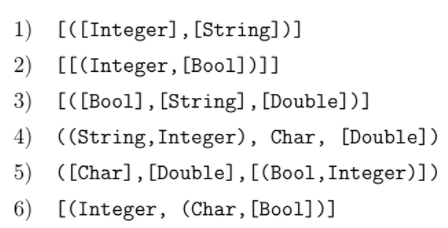


Рисунок 1 – Завдання роботи

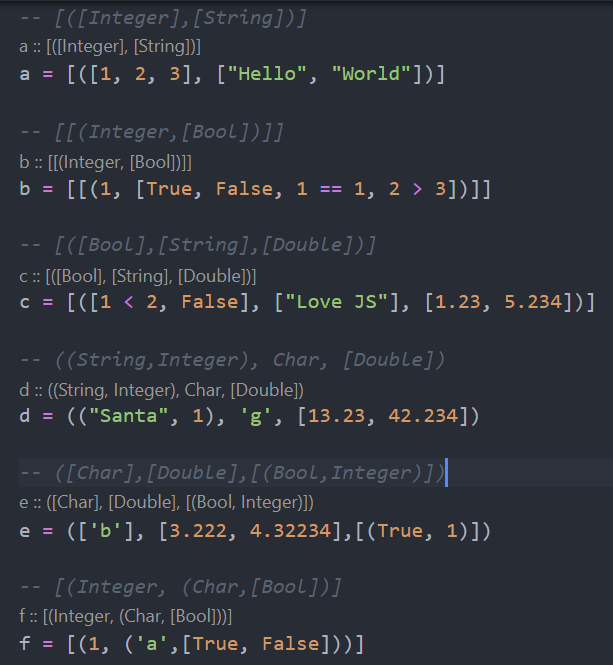


Рисунок 2 – Результати до першого завдання

1. Визначте два варiанти вказаних далi функцiй. Перший варiант – з одним аргументом-кортежем, другий – без використання кортежiв чи спискiв.

1) Функцiя приймає три числа i перевiряє, чи значення першого з них

знаходиться мiж значеннями двох iнших.

2) Функцiя за довжиною трьох вiдрiзкiв визначає, чи можна на них по-

будувати прямокутний трикутник.

3) Функцiя приймає двi логiчнi величини (Bool) i повертає їх у формi

впорядкованої за спаданням двiйки (кортежа).

4) Функцiя приймає два рядки (String) i перевiряє, чи вони лексикогра-

фiчно впорядкованi.

**Лістінг програми**

|  |
| --- |
| checkIfBetween :: [Integer] -> Bool  checkIfBetween x =  x !! 1 <= head x && head x <=  last x  checkIfBetweenOrder :: Integer -> Integer -> Integer -> Bool  checkIfBetweenOrder x y z =  y <= x && x <=  z  checkSidesForRightAngledTiangleList :: [Double] -> Bool  checkSidesForRightAngledTiangleList list = (length list == 3)                                                  && (head list^2 + list !! 1 ^2 == last list^2)                                                  || (head list^2 + last list^2 == list !! 1^2)                                                  || (list !! 1^2 + last list^2 == head list^2)  checkIfCanBuildTriangleOrder :: Integer -> Integer -> Integer -> Bool  checkIfCanBuildTriangleOrder x y z = x^2 + y^2 == z^2 || z^2 + y^2 == x^2 || x^2 + z^2 == y^2  sortLogic :: [Bool] -> (Bool, Bool)  sortLogic x = if head x > last x then (head x, last x) else (last x, head x)  sortLogicOrder :: Bool -> Bool -> (Bool, Bool)  sortLogicOrder x y = if x > y then (x, y) else (y, x)  checkIfLeksicalSortedList :: [String] -> Bool  checkIfLeksicalSortedList x = length x == 2 && head x < last x  checkIfLeksicalSorted :: (String, String) -> Ordering  checkIfLeksicalSorted (x, y) = compare x  y  checkIfLeksicalOrder :: (String, String) -> Ordering  checkIfLeksicalOrder (x, y) = compare x  y |

Таблиця 1 – Результати роботи програми з другого завдання

|  |  |  |
| --- | --- | --- |
| Функція | Тестові дані | Результати |
| checkIfBetween | 1, 0, 2 | True |
| checkIfBetweenOrder | 2, -1, 4 | True |
| checkIfCanBuildTriangle | 3, 4, 5 | True |
| checkIfCanBuildTriangleOrder | 3, 7, 2 | False |
| sortLogic | True, False | (True, False) |
| sortLogicOrder | False, True | (False, True) |
| checkIfLeksicalSorted | “a”, “b” | True (LT) |
| checkIfLeksicalSortedOrder | “b”, “a” | False (GT) |

**Висновок**

На лабораторній роботі ми ознайомилися з основними типами даних мови Haskell, зі структурою, синтаксисом та деякими функціями Glasgow Haskell Compiller. Також ми набули навичок в роботі з інтерпретатором ghci та далі запустили декілька визначених нами функцій: перевірка належності числа інтервалу, лексикографічний порядок і т. п.