

**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 інтерпретатором ghci»

Варіант 4

Виконав студент групи КН-31

Кучерук Владислав

Київ – 2021

Мета роботи

Ознайомитись з основними типами мови. Ознайомитись зі структурою та функціями Glasgow Haskell Compiler. Набути навичок роботи з інтерпретатором ghci та визначення найпростіших функцій.

Завдання

1. Наведіть приклади виразів вказаного типу. Кожен список має містити кілька елементів. Перегляньте тип прикладів, як їх визначає ghci. Прокоментуйте.

- 1) `[([Integer],[String])]`
- 2) `[[(Integer,[Bool])]]`
- 3) `[([Bool],[String],[Double])]`
- 4) `((String,Integer), Char, [Double])`
- 5) `([Char],[Double],[(Bool,Integer)])`
- 6) `[(Integer, (Char,[Bool]))]`

Рисунок 1 – Завдання роботи

```

-- [([Integer],[String])]
a :: [[Integer], [String]]
a = [[1, 2, 3], ["Hello", "World"]]

-- [([Integer],[Bool])]
b :: [[Integer], [Bool]]
b = [[(1, [True, False, 1 == 1, 2 > 3])]]

-- [([Bool],[String],[Double])]
c :: [[Bool], [String], [Double]]
c = [[(1 < 2, False), ["Love JS"], [1.23, 5.234])]

-- ((String,Integer), Char, [Double])
d :: ((String, Integer), Char, [Double])
d = (("Santa", 1), 'g', [13.23, 42.234])

-- ([Char],[Double],[ (Bool,Integer) ])
e :: ([Char], [Double], [(Bool, Integer)])
e = (['b'], [3.222, 4.32234], [(True, 1)])

-- ([Integer, (Char,[Bool])])
f :: ([Integer, (Char, [Bool])])
f = [(1, ('a', [True, False]))]

```

Рисунок 2 – Результати до першого завдання

2. Визначте два варіанти вказаних далі функцій. Перший варіант – з одним аргументом-кортежем, другий – без використання кортежів чи списків.
 - 1) Функція приймає три числа і перевіряє, чи значення першого з них знаходиться між значеннями двох інших.

- 2) Функція за довжиною трьох відрізків визначає, чи можна на них побудувати прямокутний трикутник.
- 3) Функція приймає дві логічні величини (Bool) і повертає їх у формі впорядкованої за спаданням двійки (кортежа).
- 4) Функція приймає два рядки (String) і перевіряє, чи вони лексикографічно впорядковані.

Лістинг програми

```
checkIfBetween :: (Integer, Integer, Integer) -> Bool
checkIfBetween (x, y, z) = y <= x && x <= z

checkIfBetweenOrder :: Integer -> Integer -> Integer -> Bool
checkIfBetweenOrder x y z = y <= x && x <= z

checkIfCanBuildTriangle :: (Integer, Integer, Integer) -> Bool
checkIfCanBuildTriangle (x, y, z) = x + y > z && z + y > x && x + z > y

checkIfCanBuildTriangleOrder :: Integer -> Integer -> Integer -> Bool
checkIfCanBuildTriangleOrder x y z = x + y > z && z + y > x && x + z > y

sortLogic :: (Bool, Bool) -> (Bool, Bool)
sortLogic (x, y) = if x > y then (x, y) else (y, x)

sortLogicOrder :: Bool -> Bool -> (Bool, Bool)
sortLogicOrder x y = if x > y then (x, y) else (y, x)

checkIfLeksicalSorted :: (String, String) -> Ordering
checkIfLeksicalSorted (x, y) = compare x y

checkIfLeksicalSortedOrder :: String -> String -> Ordering
checkIfLeksicalSortedOrder = compare
```

Таблиця 1 – Результати роботи програми з другого завдання

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

Висновок

На лабораторній роботі ми ознайомилися з основними типами даних мови Haskell, зі структурою, синтаксисом та деякими функціями Glasgow Haskell Compiler. 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.