**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 No9**

Виконав студент

групи АнД-31:

Пивовар Володимир Вікторович

Київ 2021

**Мета роботи**

Ознайомитись з основними типами мови. Ознайомитись зі структурою та функціями Glasgow Haskell Compiller. Набути навичок роботи з iнтерпретатором 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. Про-коментуйте.



Рисунок 1 – Заданий тип для виразу;

***Виконання 1-го завдання:***

Prelude> :set +t

Prelude> [(2.7, (True, 'a'), 5), (0.2,(False,'1'),5), (0.0,(False,'!'),8)]

[(2.7,(True,'a'),5),(0.2,(False,'1'),5),(0.0,(False,'!'),8)]

it :: (Fractional a, Num c) => [(a, (Bool, Char), c)]

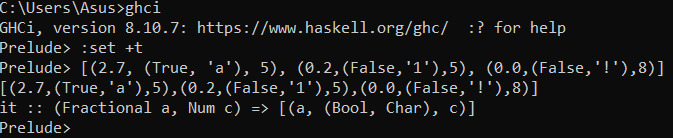


Рисунок 2 – Виконання першого завдання;

Для першого завдання було підібрано список виразів, які відповідають заданим типоп даних: [(2.7, (True, 'a'), 5), (0.2, (False, '1'), 5), (0.0, (False, '!'), 5+3)]. На першому та останньому місці стоять числа типу Double та Integer відповідно(з плаваючою комою та ціле). Для того щоб отримати тип Bool потрібно ввести True, або False, а для Char необхідно записати будб-який символ (цифру, букву, або інший символ, як ‘!’ в третьому прикладі).

**Завдання 2**

Написати функцію, що приймає три числа i повертає найменше з них (з використанням кортежу і без).

***Код прогарами:***

func1 :: Double -> Double -> Double -> Double

func1 x y z = min x (min y z)

func2 :: (Double, Double, Double) -> Double

func2 (x,y,z) = min x (min y z)

Функція func1 приймає три числа типу Double, а func2 – кортеж з трьома значеннями типу Double. Якщо спробувати передати у func1 такі ж дані, але записані у кортежі, або навпаки, у func2 просто 3 числа, то виникне помилка. Приклад роботи програми:

Prelude> :load lr1.hs

[1 of 1] Compiling Main ( lr1.hs, interpreted )

Ok, one module loaded.

\*Main> func1 2 1.1 3

1.1

\*Main> func2 (2,1.1,3)

1.1

\*Main> func1 (2,1.1,3)

<interactive>:4:7: error:

\* Couldn't match expected type `Double'

with actual type `(a0, b0, c0)'

\* In the first argument of `func1', namely `(2, 1.1, 3)'

In the expression: func1 (2, 1.1, 3)

In an equation for `it': it = func1 (2, 1.1, 3)

\*Main> func2 2 1.1 3

<interactive>:5:1: error:

\* Couldn't match expected type `t0 -> t1 -> t'

with actual type `Double'

\* The function `func2' is applied to three arguments,

but its type `(Double, Double, Double) -> Double' has only one

In the expression: func2 2 1.1 3

In an equation for `it': it = func2 2 1.1 3

\* Relevant bindings include it :: t (bound at <interactive>:5:1)

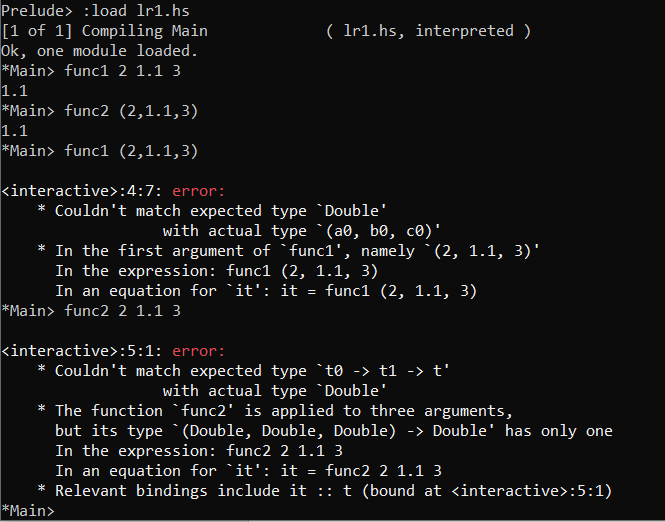


Рисунок 3 – Приклад роботи (завдання 2);

**Висновок:** У ході даної лабораторної роботи ми познайомилися з мовою програмування Haskell, встановили необхідні інструменти та написали перші програми. Виконуючи перше завдання детальніше розібралися з типами даних та написали кілька виразів за заданим типом. У другому завданні навчилися створювати функції, що приймають на вхід як кортежі, так і просто різні значення.