

# 1 Лабораторна робота № 1

## Мова Haskell. Робота з інтерпретатором ghci

### 1.1 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.

### 1.2 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]))]`
- 7) `(String,([Bool],[Integer]))`
- 8) `((Integer,Char),[Char],[String])`
- 9) `[(Double,(Bool,Char),Integer)]`
- 10) `(([Double],[Char]),[Integer])`
- 11) `[([Char],[Integer])]`
- 12) `[([Bool,String]),[Double]]`
- 13) `([Bool],[[Double],[Char,Integer]])`
- 14) `[([Double],[Bool,Char],Integer)]`
- 15) `[([String],[Bool,Integer]])]`

2. Визначте два варіанти вказаних далі функцій. Перший варіант – з одним аргументом-кортежем, другий – без використання кортежів чи списків, див. "Неформальний всуп.pdf", стор.14.

- 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.
- 5) Функція визначає, чи належить колу дана точка. Точка задається координатами, круг – координатами центра та радіусом.
- 6) Функція визначає, чи одне коло повністю міститься в іншому. Кожне коло задається координатами центра та радіусом.
- 7) Функція приймає два числа і перевіряє, чи вони впорядковані за зростанням.
- 8) Функція за довжиною трьох відрізків визначає, чи можна на них побудувати трикутник.
- 9) Функція визначає, чи перетинаються два кола. Кожне коло задається координатами центра та радіусом.
- 10) Функція приймає три числа і повертає найменше з них.
- 11) Функція визначає, чи містить круг початок координат. Круг задається координатами центра та радіусом.
- 12) Функція визначає, чи дві точки, задані координатами, знаходяться у одному квадранті.
- 13) Функція приймає три рядки (String) і повертає лексикографічно найбільше з них.
- 14) Функція за довжиною чотирьох відрізків визначає, чи можна на них побудувати прямокутник.
- 15) Функція приймає два символи (Char) і повертає їх у формі впорядкованої за зростанням двійки (кортежа).

### 1.3 Порядок виконання лабораторної роботи № 1

Лабораторна робота передбачає виконання двох завдань – 1.N та 2.N, де N – номер студента у списку підгрупи.

Нехай номер у списку підгрупи – 1. Тоді треба виконати завдання 1.1 та 2.1 у двох варіантах – 2.1a та 2.1б.

Після виконання завдань та оформлення звіту, лабораторна робота захищається. Захист зводиться до демонстрації роботи програм, пояснення результатів їх виконання, модифікації програм (за ініціативи викладача), та відповіді на контрольні питання.

### 1.4 Звіт про виконання лабораторної роботи

Вимоги – звичайні:

1. формат – pdf;
2. титульна сторінка (у т.ч. тема, номер варіанта);
3. мета виконання роботи;

4. для кожного завдання:
  - 1) номер;
  - 2) повний текст завдання;
  - 3) код;
  - 4) протокол тестування (текстовий формат, НЕ скриншоти);
  - 5) коментар та/чи пояснення;
5. загальний висновок (принаймні, враження).

## 1.5 Питання для самоконтролю

1. Мова Haskell:
  - 1) загальна характеристика;
  - 2) основні типи;
  - 3) функції для роботи з основними типами;
  - 4) правила найменування змінних, функцій, типів;
  - 5) зіставлення зі зразком;
  - 6) умовні вирази;
  - 7) визначення функцій.
2. Загальна характеристика GHC.
3. Інтерпретатор ghci:
  - 1) загальна характеристика;
  - 2) команди для роботи з типами;
  - 3) команди для роботи з файлами програм.

## 1.6 Корисні посилання

1. <https://www.haskell.org/>
2. Introductory Books for Learning Haskell, <https://www.haskell.org/documentation/>
3. Неформальний вступ.Haskell.ghci.pdf