

# Лабораторна робота № 4

## з дисципліни

### ”Функційне програмування мовою Haskell”

## 1 Робота № 4

### Розробка прикладних програм мовою Haskell

#### 1.1 Мета роботи

Ознайомитись з модульною організацією програм та засобами введення-виведення. Набути досвіду створення мультимодульних Haskell-проектів з використанням інструментів `stack` або `cabal`.

#### 1.2 Завдання

За допомогою інструментів `stack` або `cabal` створити Haskell-проект для розробки консольного застосунку за темою попередньої лабораторної роботи (ЛР № 3).

Реалізувати інтерфейс командного рядка: а) для виконання функцій, реалізованих у ЛР № 3, б) для маніпуляції даними (CRUD).

Дані зберігаються або у текстовому файлі (файлах), або у csv-файлі (файлах) — на вибір студента.

Звіт (результат виконання) виводиться а) на екран (якщо не заборонено, тобто не вказана опція `[-s|--silent]`), б) у текстовий файл, в) у html-файл.

Інтерфейс має підтримувати набір непозиційних параметрів (прапорців) з, наприклад, таким форматом:

```
programName [DBNAME] [-c|--command COMMANDNAME [ARGS]]
                  [-l|--log LOGNAME]
                  [-s|--silent]
                  [--html FILENAME]
                  [-h|--help]
```

Наприклад, для теми "записник" запит на перегляд довідки міг би виглядати так:

```
sketchbook -h
```

### 1.3 Теми для обговорення

1. Монади.
2. Монадичне зв'язування.
3. `do`-нотація.
4. Монада `IO`.
5. Інші монадичні типи модуля `Prelude`.
6. Компілятор `ghc`.
7. `Stack`.
8. `Cabal`.

### 1.4 Корисні посилання

1. <https://www.haskell.org/>
2. Introductory Books for Learning Haskell. <https://www.haskell.org/documentation/>
3. <http://learnyouahaskell.com/chapters>
4. <http://learnyouahaskell.com/input-and-output>
5. <http://learnyouahaskell.com/for-a-few-monads-more>
6. <https://book.realworldhaskell.org/read/>
7. <https://hackage.haskell.org/>