

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені  
ТАРАСА ШЕВЧЕНКА**



**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №4**

**з курсу**

**«Функціональне програмування»**

*Студентки 4 курсу  
групи ПП-41*

*спеціальності 122 «Комп'ютерні науки»*

*ОП «Прикладне програмування»*

*Штось Софії Максимівни*

*Викладач:  
Пирог М. В.*

**Київ – 2023**

**Тема роботи:** Часткове використання функцій

**Мета роботи:** Ознайомитися з відмінностями між каррінгом та частковим використанням функцій. Розширити навички каррінгу та часткового використання функцій. Поглибити знання в області частково параметризованих функцій.

### ***Теоретичні відомості***

Тема №4 «Часткова параметризація функцій» має на меті розкриття основних аспектів написання програм з використанням частково параметризованих функцій. Для цього описуються особливості каррінгу, розглядається функціональне представлення даних та позиціонування у списках. На самостійне опрацювання виноситься питання часткового використання функцій, оскільки каррінг часто плутають з частковим використанням і необхідно розуміти різницю.

Для поглиблення знань в області частково параметризованих функцій, а саме в частковому використанні функцій слід особливу увагу звернути на різницю та спільні риси понять каррінг та часткове використання функцій. Необхідно розглянути Часткові (обмежені) функції та механізм реалізації часткового використання функцій в різних мовах програмування та математичні основи цього процесу. Слід також приділити увагу сферам практичного програмування, де застосовується каррінг та часткове використання функцій та розглянути типові користувацькі сценарії, де використовується кожне з явищ.

### ***Завдання для виконання***

Реалізувати функцію обчислення добутку п'яти натуральних чисел. Застосувати часткове використання функції до 1-го, 2-х, 3-х та 4-х аргументів, які задати за замовчуванням.

### ***Хід роботи***

*Код програми (Haskell):*

```
module Main (main) where

getProduct :: Int -> (Int -> (Int -> (Int -> (Int -> Int))))
getProduct = \a -> \b -> \c -> \d -> \e -> a * b * c * d * e

main :: IO ()
```

```

main = do

  let partial1 = getProduct 2

  let partial2 = partial1 3

  let partial3 = partial2 4

  let partial4 = partial3 5


  let result = partial4 6


  putStrLn $ "Добуток п'яти натуральних чисел: " ++ show result

```

*Результат роботи програми:*

The screenshot shows a Haskell IDE with two panels. The top panel displays the source code for `Main.hs`, which defines a `getProduct` function and a `main` function that calculates the product of five numbers (2, 3, 4, 5, 6) using partial application. The bottom panel shows the terminal output, which includes the compilation and linking process for `Lab4-exe` and the final output: `Добуток п'яти натуральних чисел: 720`.

```

Main.hs
app > Main.hs > Main > main
1  module Main (main) where
2
3  getProduct :: Int -> (Int -> (Int -> (Int -> (Int -> Int))))
4  getProduct = \a -> \b -> \c -> \d -> \e -> a * b * c * d * e
5
6  main :: IO ()
7  main = do
8      let partial1 = getProduct 2
9      let partial2 = partial1 3
10     let partial3 = partial2 4
11     let partial4 = partial3 5
12
13     let result = partial4 6
14
15     putStrLn $ "Добуток п'яти натуральних чисел: " ++ show result

```

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL
> v TERMINAL
Configuring Lab4-0.1.0.0...
Lab4> build (lib + exe)
Preprocessing library for Lab4-0.1.0.0..
Building library for Lab4-0.1.0.0..
Preprocessing executable 'Lab4-exe' for Lab4-0.1.0.0..
Building executable 'Lab4-exe' for Lab4-0.1.0.0..
[1 of 2] Compiling Main [Source file changed]
[3 of 3] Linking .stack-work/dist/aarch64-osx/ghc-9.4.7/build/Lab4-exe/Lab4-exe [O
Lab4> copy/register
Installing library in /Users/sofiiashton/Documents/GitHub/func-prog-shton/haskell-
9443c89/9.4.7/lib/aarch64-osx-ghc-9.4.7/Lab4-0.1.0.0-2f0S1cxRji02vk3y1o2ZWg
Installing executable Lab4-exe in /Users/sofiiashton/Documents/GitHub/func-prog-sh
7a65f338d6789443c89/9.4.7/bin
Registering library for Lab4-0.1.0.0..
Добуток п'яти натуральних чисел: 720

```

**Висновок:** Отже, у ході цієї лабораторної роботи було проведено ознайомлення з каррінгом та частковим використанням функцій. Було досліджено їх особливості та відмінності, а також було набуто практичні навички з їх використання.

**Контрольні запитання**

## *1. Які спільні та відмінні риси мають каррінг та часткове застосування функцій мовою Haskell?*

Каррінг і часткове застосування функцій, хоч і є концепціями функціонального програмування, мають свої відмінності. Обидві ідеї дозволяють розділяти параметри функцій, але способи їх використання та підходи трошки різняться. Наприклад, каррінг дозволяє представляти функції з декількома параметрами як послідовність функцій, кожна з яких має лише один параметр. У Haskell функції і так можна викликати з частковою передачею параметрів, але це не обов'язково пов'язано з каррінгом. Також, каррінг частіше використовується для створення нових функцій шляхом часткового застосування, тоді як часткове застосування в Haskell може використовуватися більш широко в контексті часткового передавання параметрів.

## *2. В чому полягають особливості застосування часткового використання функцій?*

Серед ключових особливостей часткового використання функцій можна виділити наступні:

- *Часткове передавання параметрів.* Замість передачі всіх параметрів одним махом, можна передавати їх поетапно. Це дозволяє створювати нові функції, які мають частково визначені параметри.
- *Створення нових функцій.* Часткове застосування дозволяє створювати нові функції на основі існуючих, фіксуючи певні параметри і залишаючи інші для подальшого визначення.
- *Зручність та гнучкість.* Це дозволяє писати більш зручний та гнучкий код, особливо в ситуаціях, де часткова передача параметрів забезпечує повторне використання коду.
- *Каррінг.* Часткове застосування функцій часто пов'язане з концепцією каррінгу, де функція може бути викликана послідовно з частково переданими параметрами.
- *Лямбда-вирази.* Мови програмування, які підтримують лямбда-вирази, можуть легше виражати часткове застосування функцій, що полегшує роботу з ним.

- *Функції вищого порядку.* Часткове застосування сприяє використанню функцій вищого порядку, оскільки дозволяє передавати функції як аргументи і повертати їх як результати.

### *3. Який зв'язок мають явища каррінг та часткове використання функції?*

По-перше, каррінг дозволяє розглядати функцію з кількома аргументами як послідовність однопараметричних функцій. Це відкриває можливості для часткового використання, коли передаються лише окремі параметри, створюючи нові функції. Також, використання часткової функції можна розглядати як використання каррінгу. Створення нових функцій шляхом фіксації окремих параметрів відбувається відповідно до принципів каррінгу.

### *4. Яким чином реалізується часткове використання функцій мовою програмування Haskell?*

У Haskell часткове використання функцій можна реалізувати декількома способами, завдяки функціональній природі мови. Цими способами є каррінг, використання лямбда-виразів, та функції вищого порядку.

### *5. У яких випадках застосовується часткове використання функцій при вирішенні практичних задач?*

Часткове використання функцій є корисним в різних практичних сценаріях програмування. Зокрема, це сприяє зменшенню кількості рядків коду, оскільки дозволяє фіксувати параметри та створювати нові функції без повторення коду. Це також сприяє гнучкості, оскільки частково застосовані функції можна легко модифікувати, передаючи їм частковий набір параметрів. Крім того, вони ефективно взаємодіють із функціями вищого порядку та спрощують композицію функцій. Цей підхід також допомагає оптимізувати код, уникати дублювання та полегшує роботу з функціями у різних сферах програмування.