

## ЛАБОРАТОРНА РОБОТА № 3

### Реалізація інструкцій вибору й ітерації C/C++ у модулях та програмних засобах

*Мета роботи* — набути практичних навичок застосування операторів умови if, циклів for, while, do...while та вибору switch мови програмування C/C++ під час розроблення функцій статичних бібліотек і модульного програмного забезпечення у кросплатформовому Code::Blocks IDE.

#### ЧАС ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

4 академічні години.

#### ОБЛАДНАННЯ, МАТЕРІАЛИ, ПРОГРАМНІ ЗАСОБИ

Для виконання лабораторної роботи необхідні:

- персональний комп'ютер з ОС Windows XP / Vista / 7 / 8.x / 10, Linux 32-bit / 64-bit або Mac OS X;
- вільне кросплатформове середовище розроблення програмного забезпечення Code::Blocks ([www.codeblocks.org](http://www.codeblocks.org)) для відповідної платформи: Windows XP / Vista / 7 / 8.x / 10, Linux 32-bit / 64-bit, або Mac OS X;
- текстовий редактор (OpenOffice Writer, Microsoft Word або ін.).

#### Завдання до лабораторної роботи

1. Реалізувати модульні функції розв'язування задач 3.1-3.3 як складових статичної бібліотеки *ПрізвищеLib*, створеної під час виконання лабораторної роботи № 2 (`\MP\Прізвище_Lab2\MyLib\`).
2. Реалізувати модульне програмне забезпечення (програмний засіб) розв'язування задачі 3.4.

### Порядок виконання лабораторної роботи ТА МЕТОДИЧНІ ВКАЗІВКИ

1. У викладача отримати умови задач 3.1-3.4 за варіантом.
2. До звіту з лабораторної роботи (далі — звіт) записати тему й мету роботи, номер варіанту, завдання.
3. Почергово здійснити аналіз задач 3.1, 3.2 та 3.3.
4. Розробити три набори контрольних прикладів до задач 3.1-3.3 відповідно задля проведення модульного тестування (Unit Testing); належно їх задокументувати та включити до звіту.
5. Почергово здійснити функціональну декомпозицію задач 3.1-3.3, детальне проектування – формалізацію (алгоритмізацію); одержані артефакти задокументувати й включити до звіту.
6. В Code::Blocks IDE відкрити проект статичної бібліотеки *ПрізвищеLib* (`\MP\Прізвище_Lab2\MyLib\`), створеної під час виконання лабораторної роботи № 2.
7. За артефактами процесу детального проектування (п. 5) здійснити конструювання модульних функцій: мовою програмування C або C++ описати належно іменовані функції, які за інтерфейсом реалізують розв'язування задач 3.1, 3.2 та 3.3 відповідно.
8. Скомпілювати проект статичної бібліотеки *ПрізвищеLib*, в результаті чого з файлу *ПрізвищеLib.cpp* має створитись файл *ПрізвищеLib.a*.
9. Відкрити проект заголовкового файлу *ПрізвищеModule*, створений під час виконання лабораторної роботи № 2 (у `\MP\Прізвище_Lab2\MyLib\`), та додати прототипи розроблених функцій задач 3.1, 3.2 й 3.3 (скопювати з *ПрізвищеLib.cpp* заголовки функцій).
10. В Code::Blocks IDE розробити тестовий(і) драйвер(и) для розроблених трьох тест-сьютів і здійснити модульне тестування функцій задач 3.1, 3.2 та 3.3 (для підключення/включення функцій бібліотеки до файлу програми слід

використати директиву препроцесора `#include "ПрізвищеModule.h"` й налаштувати опції дебагера `Build options...` : `Linker` – шлях до файла статичної бібліотеки `ПрізвищеLib.a`, `Compiler` – шлях до її заголовкового файла `ПрізвищеModule.h`).

- у випадку невиконання тест-кейса(ів) слід здійснити відлагодження відповідного модуля (зазвичай, з повторним виконанням детального проектування й конструювання), після чого процес тестування повторити.
11. Артефакти процесу модульного тестування задокументувати шляхом включення результатів роботи тестового драйвера до звіту (наприклад, скопіювати текст з консольного вікна Windows у буфер ОС можна наступним чином: контекстне меню → Виділити все, після чого натиснути <Enter>);
  12. Здійснити аналіз задачі 3.4 та вимог до ПЗ;
  13. Виконати функціональну декомпозицію задачі 3.4, проектування архітектури й детальне проектування програмного засобу; отримані артефакти задокументувати, включити до звіту.
  14. Розробити тестовий набір (Test Suite) для здійснення системного тестування (System Testing) ПЗ задачі 2.3, який повинен складатись з достатньої кількості тест-кейсів (є допустимим використання простої структури тест-кейсів: Action (test steps) → Expected Result → Test Result (passed/failed/blocked); запропонований тест-сьют включити до звіту.
  15. В Code::Blocks IDE створити проект консольного додатка `Прізвище_Lab3` у теці `\MP\Прізвище_Lab3\prg\`.
  16. Мовою програмування C/C++ реалізувати програмне забезпечення за артефактами процесів проектування програмного засобу задачі 3.4 (для використання модульних функцій задач 3.1-3.3 слід використати директиву препроцесора `#include "ПрізвищеModule.h"`).
  17. Здійснити системне тестування розробленого програмного засобу `Прізвище_Lab3.exe` за допомогою створеного тестового набору (п. 14);

- у випадку невиконання тест-кейса(ів) слід виконати повторне проектування й конструювання програмного забезпечення, після чого тестування програмного засобу повторити.

18. Результати системного тестування програмного забезпечення `Прізвище_Lab3.exe` задокументувати й включити до звіту.
19. Розроблений додаток `Прізвище_Lab3.exe` скопіювати у `\MP\Прізвище_Lab3\my_software\`.
20. Затвердити у викладача реалізоване програмне забезпечення (exe-файл у `\MP\Прізвище_Lab3\my_software\`).
21. Лістинги розроблених функцій статичної бібліотеки (файл `ПрізвищеLib.cpp`) та програмного засобу (`Прізвище_Lab3.cpp`) включити до звіту.
22. Одержані результати виконання завдань зберегти на носій (флеш-накопичувач, хмарне сховище даних тощо) з метою забезпечення можливості їх подальшого використання під час виконання наступних лабораторних робіт.
23. Проаналізувати хід виконання лабораторної роботи та одержані результати, сформулювати обґрунтовані висновки (підсумки)<sup>1</sup> обсягом не менше ½ сторінки машинописного тексту та включити їх до звіту; у висновках варто також окремо зазначити особисті враження від процесу виконання завдань, викласти вмотивовані пропозиції, обґрунтовані рекомендації, зауваження, конструктивну критику<sup>2</sup> тощо.
24. Підготувати звіт з лабораторної роботи відповідно до встановлених вимог щодо його структури, змісту, оформлення.
25. Подати викладачу звіт до захисту.

<sup>1</sup> висновки, як результат розумової діяльності студента, повинні, зокрема, містити стисле викладення здобутих в процесі виконання лабораторної роботи результатів, реалізованих ідей, опис проблем, які виникали під час реалізації завдань, та шляхи їх вирішення; структура підсумків має бути логічною і охоплювати весь процес виконання лабораторної роботи тощо.

<sup>2</sup> критика є розглядом і оцінкою когось чи чогось з метою виявлення й усунення вад, хиб; під конструктивною слід розуміти критику, після якої стає зрозумілим, як саме виправити помилку й не допускати її в майбутньому.

### КОНТРОЛЬНІ ЗАПИТАННЯ І ЗАВДАННЯ

1. Яким оператором C/C++ можливо повноцінно замінити тернарний оператор? Відповідь обґрунтуйте й доведіть експериментально.
2. Що в програмуванні розуміють під пріоритетом виконання операцій та асоціативністю?
3. Яку область видимості мають змінні, оголошені в тілі циклу або умови (вибору)? Відповідь обґрунтуйте та доведіть експериментально.
4. Якою є асоціативність операцій арифметичних, логічних, логічних порозрядних, інкрементна, декрементна, тернарної та порівняння в мові програмування C/C++?
5. Перелічіть випадки, за яких доцільно використовувати тернарний оператор C/C++, й наведіть приклад його запису.
6. Яке значення міститиме змінна `cnt` після виконання наступної інструкції: `cnt--;` ?
7. Чим константна змінна, оголошена за допомогою кваліфікатора типів `const`, відрізняється від змінної? Сформулюйте правило, коли змінну варто оголошувати саме константною.
8. Яких типів можуть бути операнди логічних операторів C/C++?
9. Яке значення міститиме змінна `cnt` при: `bool cnt = !!0;` ?
10. Сформулюйте правило запису виразу ініціалізації у циклах з параметром (`for`) C++.
11. Наведіть приклад запису циклу `for`, який виконується до введення користувачем символу «!» без використання жодної змінної, оголошеної до цього циклу.
12. Яким є синтаксис запису оператора повного й неповного розгалуження у C/C++? Перелічіть допустимі та недопустимі операції у логічних виразах оператора умови.
13. Які логічні операції реалізовано у C/C++? Перелічіть логічні оператори C/C++, типи їх операндів і результату виконання.

14. Яке значення міститиметься у змінній `cnt` після `++cnt-=1;` ?
15. Наведіть приклад опису константної змінної в модулі C/C++.
16. Яке призначення має оператор `break` у C/C++ та в чому полягає особливість його виконання?
17. Які логічні порозрядні операції реалізовано у C/C++? Перелічіть унарні й бінарні оператори порозрядних операцій, типи їх операндів та результату виконання? Наведіть приклади.
18. Сформулюйте критерії алгоритмічної конструкції повторення (ітерації), за якими слід здійснювати вибір конкретного виду цикла (`for`, `while` або `do...while`) для їх опису мовою програмування C/C++.
19. Сформулюйте випадки й ситуації в алгоритмах, опис яких мовою програмування варто здійснити за допомогою оператора `continue`.
20. В чому полягає сутність методики створення статичних бібліотек у Code::Blocks IDE?