

3. Перелічіть та поясніть стадії утворення модульного ПЗ (процес трансляції .cpp в .exe)?
4. Які слова у C/C++ називаються ключовими (escape-последовності) і яке їх призначення?
5. Яке призначення директиви препроцесора #include у C/C++?
6. Що міститься у заголовковому файлі?
7. Яким є синтаксис запису числових, символьних, рядкових літералів (літеральних констант) у C/C++?
8. Сформулюйте правило побудови ідентифікаторів (в тому числі щодо його довжини, регістра) C/C++.
9. Перелічіть вбудовані числові й символьні типи даних C/C++, їх розмірність (в байтах), діапазон.
10. Яким є синтаксис оголошення змінних у C/C++? Чи можливо одночасно оголосити й ініціалізувати змінну за допомогою константного літералу або виразу?
11. Що розуміють під інтерфейсом функції або модуля і який синтаксис його запису мовою програмування C/C++?
12. За допомогою якої функції заголовкового файлу clocale/locale.h здійснюють локалізацію кодування символів?
13. Яким є порядок створення статичної бібліотеки у Code::Blocks IDE та використання функцій з неї?
14. У якому заголовковому файлі стандартної бібліотеки визначені математичні функції мови програмування C, а у якому – C++?
15. Яким чином у C++ реалізовується потокове виведення даних з та без форматування?
16. За допомогою якого об'єкта C++ здійснюється введення даних?
17. Яке призначення маніпуляторів boolalpha, hex, fixed, endl?
18. Поясніть сутність і призначення кожного етапу побудови модульного ПЗ: препроцесинг (preprocessing), компіляція (compiling), зв'язування (linking).
19. Яким чином реалізовується тестування модуля (Unit testing)?
20. В межах якого процесу життєвого циклу ПЗ розробляється набір контрольних прикладів (тест-кейсів)?

ЛАБОРАТОРНА РОБОТА № 2

РЕАЛІЗАЦІЯ ВВЕДЕННЯ Й ВИВЕДЕННЯ ДАНИХ ТА МАТЕМАТИЧНИХ ФУНКЦІЙ C/C++

Мета роботи — набуття практичних навичок функціональної декомпозиції обчислювальних задач і процесів, опису й використання змінних, застосування об'єктів потокового введення та виведення даних, стандартних математичних функцій C/C++, реалізації власних модулів і програмного забезпечення у вільному кросплатформовому середовищі Code::Blocks (GNU GCC Compiler).

ЧАС ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

4 академічні години.

ОБЛАДНАННЯ, МАТЕРІАЛИ, ПРОГРАМНІ ЗАСОБИ

Для виконання лабораторної роботи необхідні:

- персональний комп'ютер з операційною системою Windows;
- вільне кросплатформове середовище розроблення програмного забезпечення Code::Blocks (www.codeblocks.org);
- текстовий редактор (OpenOffice Writer, Microsoft Word або ін.).

Завдання до лабораторної роботи

1. Реалізувати модуль з функціями розв'язування задач 2.1 та 2.2.
2. Реалізувати модульне програмне забезпечення (консольний додаток для ОС Windows) розв'язування задачі 2.3.

ПОРЯДОК ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ ТА МЕТОДИЧНІ ВКАЗІВКИ

1. У викладача отримати умови задач 2.1-2.3 за варіантом.
2. До звіту з лабораторної роботи (далі — звіт) записати тему й мету роботи, номер варіанту, завдання.

3. Здійснити аналіз задач 2.1 та 2.2, вимог до ПЗ.
4. Розробити набори контрольних прикладів до задач 2.1 та 2.2 задля проведення модульного тестування (Unit testing), належно їх задокументувати, включити до звіту.
5. Здійснити функціональну декомпозицію задач 2.1 та 2.2 і детальне проектування модуля; одержані артефакти задокументувати й включити до звіту.
6. В Code::Blocks IDE створити проект статичної бібліотеки *ПрізвищеLib* у \MP\Прізвище_Lab2\MyLib\.
7. На основі результатів детального проектування модуля (п. 5) мовою програмування C або C++ описати функції *q_calculation* та *s_calculation*, які за належним інтерфейсом реалізують розв'язування задач 2.1 і 2.2 відповідно.
8. Скопіювати створений файл статичної бібліотеки *ПрізвищеLib*, в результаті чого з файлу *ПрізвищеLib.cpp* має створитись об'єктний файл *ПрізвищеLib.a*.
9. Створити проект заголовкового файлу *ПрізвищеModule* в \MP\Прізвище_Lab2\MyLib\ та описати в ньому прототипи функцій *q_calculation* і *s_calculation* (скопювати з бібліотеки *ПрізвищеLib* заголовки функцій).
10. Розробити драйвер для виконання розроблених тестових наборів і здійснити модульне тестування функцій *q_calculation* та *s_calculation* статичної бібліотеки *ПрізвищеLib* (для підключення бібліотеки слід використати директиву препроцесора `#include "ПрізвищеModule.h"` та налаштувати опції дебагера Build options...: Linker – шлях до об'єктного файлу статичної бібліотеки *ПрізвищеLib.a*, Compiler – шлях до її заголовкового файлу *ПрізвищеModule.h*); результати тестування належно задокументувати, включити до звіту.
11. У випадку негативного результату тестування модуля здійснити його відлагодження, процес тестування повторити.
12. Здійснити аналіз задачі 2.3 та вимог до ПЗ, виконати функціональну декомпозицію, проектування архітектури й

детальне проектування програмного забезпечення; отримані результати задокументувати, включити до звіту.

13. Розробити тестовий набір (Test Suite) для здійснення системного тестування (System Testing) програмного забезпечення розв'язування задачі 2.3, який повинен складатись не менш як з п'яти тестових випадків (Test Case); отриманий артефакт задокументувати, включити до звіту.
14. В Code::Blocks IDE створити проект консольного додатка *Прізвище_task* у \MP\Прізвище_Lab2\prg\.
15. Мовою програмування C/C++ реалізувати програмне забезпечення розв'язування задачі 2.3 на основі результатів проектування архітектури й детального проектування ПЗ (п. 11); для підключення й використання функцій *q_calculation* і *s_calculation* слід використати директиву препроцесора `#include "ПрізвищеModule.h"` (див. п. 10).
16. Відповідно до вимог міжнародного стандарту ISO/IEC 12207 здійснити системне тестування (System Testing) реалізованого програмного засобу *Прізвище_task.exe* за допомогою розробленого тестового набору (п. 12); результати тестування задокументувати, включити до звіту.
17. У випадку негативного результату тестування *Прізвище_task.exe* здійснити відлагодження програми, тестування повторити.
18. Реалізований додаток *Прізвище_task.exe* скопіювати у \MP\Прізвище_Lab2\my_software\.
19. Розроблене програмне забезпечення задачі 2.3 (exe-файл з теки *my_software*) затвердити у викладача.
20. Лістинги розроблених бібліотечних функцій (файл *ПрізвищеLib.cpp*) та модульного ПЗ (*Прізвище_task.cpp*) включити до звіту.
21. Одержані результати виконання лабораторної роботи зберегти на носій (флеш-накопичувач, хмарне сховище даних тощо) з метою їх подальшого використання під час виконання наступних лабораторних робіт.

22. Здійснити аналіз ходу виконання завдань лабораторної роботи й одержаних результатів.
23. Сформулювати обґрунтовані висновки (підсумки)¹ з виконаної лабораторної роботи обсягом не менше ½ сторінки машинописного тексту та включити їх до звіту; у висновках рекомендовано також окремо зазначити особисті враження, вмотивовані пропозиції, рекомендації, застереження, зауваження тощо.
24. Підготувати звіт відповідно до встановлених вимог щодо його структури, змісту, оформлення.
25. Подати викладачу звіт до захисту.

КОНТРОЛЬНІ ЗАПИТАННЯ І ЗАВДАННЯ

1. Яка мета й задачі процесів проектування програмного забезпечення відповідно до міжнародного стандарту ISO/IEC 12207:2008?
2. Обґрунтовано поясніть, чим функція мови програмування C/C++ відрізняється від модуля.
3. Яка відмінність функції main від решти функцій C/C++?
4. Яке призначення маніпуляторів і яким чином вони використовуються під час реалізації програмного забезпечення мовою програмування C++?
5. Як і для чого використовується заголовковий файл в процесі препроцесинга програми?
6. Що під час виконання лабораторної роботи Вами використано зі стандартного заголовкового файлу iostream та задля реалізації яких функцій?
7. Що розуміють під стандартним простором імен у C++ і якою директивою він визначається?

¹ 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.

8. Наведіть приклади одночасного оголошення й ініціалізації початковим значенням змінної (об'єкта).
9. Перелічіть символи, використання яких у ідентифікаторах не допускаються відповідно до синтаксису мови програмування C/C++.
10. Яким чином у C++ можливо вивести десятковий літерал у шістнадцятковій системі числення?
11. Що необхідно виконати для підключення й використання функцій нестандартної (наприклад, власної) бібліотеки?
12. Чим відрізняється заголовковий файл від об'єктного?
13. Яким чином вивести нестандартні символи (наприклад, $\sqrt{\quad}$) під час реалізації мовою програмування C або C++ консольного додатка для ОС Windows?
14. В чому полягає відмінність між записом символьного і рядкового константного літерала в C/C++?
15. Яким є синтаксис запису прототипа функції у C/C++? Де він записується у програмі та яке його призначення?
16. Який оператор C/C++ призначений для повернення функцією значення - результату і який синтаксис його запису?
17. Перелічіть ключові символи (ESC-послідовності) мови програмування C, їх призначення, синтаксис запису та спосіб використання у C++.
18. Перелічіть відомі Вам функції заголовкового файлу cmath та їх призначення? В чому полягає їх відмінність від функцій бібліотеки math.h?
19. Яке призначення тестового драйвера і яким чином він реалізовується?
20. Яким чином здійснюється тестування модулів (функцій бібліотек) і чим означений процес відрізняється від тестування програмного засобу (exe-файла)?