

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет

ЗВІТ
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 8
з навчальної дисципліни
“Базові методології та технології програмування”
РЕАЛІЗАЦІЯ СТАТИЧНИХ БІБЛІОТЕК МОДУЛІВ ЛІНІЙНИХ
ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

ЗАВДАННЯ ВИДАВ
доцент кафедри кібербезпеки
та програмного забезпечення
Доренський О. П.
<https://github.com/odorenskyi/>

ВИКОНАВ
студент академічної групи КБ-23
Чернолес К.С.

ПЕРЕВІРИВ
ст. викладач кафедри кібербезпеки
та програмного забезпечення
Дреєва Г.М.

Лабораторна робота №8

Тема: реалізація статичних бібліотек модулів лінійних обчислювальних процесів

Мета роботи: полягає у набутті ґрунтовних вмінь і практичних навичок застосування теоретичних положень методології модульного програмування, реалізації метода функціональної декомпозиції задач, метода модульного (блочного) тестування, представлення мовою програмування C++ даних скалярних типів, арифметичних і логічних операцій, потокового введення й виведення інформації, розроблення програмних модулів та засобів у кросплатформовому середовищі Code::Blocks (GNU GCC Compiler).

Завдання:

1. Реалізувати статичну бібліотеку модулів libModulesПрізвище C/C++, яка містить функцію розв'язування задачі 8.1.
2. Реалізувати програмне забезпечення розв'язування задачі 8.2 — консольний застосунок.

Варіант 2

1.

ВАРІАНТ 2

— ЗАДАЧА 8.1 —

За значеннями x, y, z обчислюється S :

$$S = \frac{\sqrt{1+x} - 3 \cos x}{\ln x^2 + 3 \sin \pi x} + \left(\sqrt{z^3} + 2 \right)^2.$$

Малюнок 1 – Умова задачі 8.1

Строго постановка задачі:

Вхідні дані: x, z – дійсні числа:

Вихідні дані: S – дійсне число.

Проектування програмного модуля:

Модуль ModulesChernoles складається із заголовкового файлу, що містить оголошення прототипу функцій, та файлу вихідного коду, в якому зберігається реалізація функції `s_calculation`.

Функція `s_calculation` приймає три аргументи в якості вхідних даних та повертає результат виразу, зазначеного в умові.

Тестовий драйвер `TestDriver` – функція, призначена для тестування функції. Вона містить:

- масив з вхідними даними;
- масив з очікуваними результатами;
- цикл для перебору наборів значень та виводу результату тестування певного набору.

Тест-сьют модульного тестування статичної бібліотеки

ModulesChernoles.a

наведений у додатку А.

Лістинги вихідного коду проєктів ModulesChernoles та TestDriver містяться в Додатку Б та Додатку В відповідно.

Результати тестування s_calculation зі статичної бібліотеки libModulesChernoles.a тестовим драйвером:

```

TestDriver of ModulesChernoles.a
TC_01: | x = 0      z = 0      | - result: - passed
TC_02: | x = 1      z = 0      | - result: - passed
TC_03: | x = 0      z = 0      | - result: - passed
TC_04: | x = 0      z = 1      | - result: - passed
TC_05: | x = -2.5   z = 0      | - result: - passed
TC_06: | x = -2.5   z = -3.12 | - result: - passed
TC_07: | x = 0      z = -2     | - result: - passed
TC_08: | x = 4.65   z = 0      | - result: - passed
TC_09: | x = 2      z = 6      | - result: - passed
TC_10: | x = -11    z = -9     | - result: - passed

```

2.

— ЗАДАЧА 8.2 —

За послідовними запитами вводяться числа x , y , z та символи a і b .

Вивести (включити у потік STL — cout)*:

8.2.1. Прізвище та ім'я розробника програми зі знаком охорони авторського права «©» (від англ. copyright);

8.2.2. Результат логічного виразу в числовому вигляді (1/0):

$$a + 7 < |b - 5| ?$$

8.2.3. Значення x , y , z в десятковій і шістнадцятковій системах числення; S , що обчислюється функцією s_calculation() заголовкового файлу ModulesПрізвище.h.

* Підзадачі 8.2.1–8.2.3 варто реалізувати у вигляді функцій, результат виконання яких включається у вихідний потік cout за допомогою оператора вставки << (наприклад, “cout << YourFunc(a,b);”).

Строго постановка задачі:

Вхідні дані: x, z – цілі числа,

a, b – символічні літерали;

Вихідні дані: прізвище та ім'я розробника з символом «©»,

результат логічного виразу,

значення x, z в десятковій та шістнадцятковій системах

числення,

S , що обчислюється функцією `s_calculation()` заголовкового файлу `ModulesChernoles.h` ;

Проектування програмного модуля:

Модуль `ModulesChernoles` використати з попередньої задачі.

На початку програма окремо запитує у користувача 2 числа, які записуються у змінні a, b . та 2 числа, які записуються в змінні x, z .

Перелічені функції викликаються у функції `main`, до того ж до потоку включається функція `s_calculation()` з модуля `ModulesChernoles`, яка приймає в якості аргументів змінні x, z та повертає результат виразу.

Результати системного тестування ПЗ `Chernoles_task.exe` зазначені у додатку Г.

Висновок: дана лабораторна робота була націлена на набуття навичок у використанні теоретичних положень модульної парадигми програмування, реалізації метода функціональної декомпозиції задач, метода модульного тестування, а також у представленні мовою програмування C++ даних скалярних типів, арифметичних і логічних операцій, потокового введення та виведення інформації, розробленні програмних модулів та засобів у кросплатформовому середовищі `Code::Blocks (GNU GCC Compiler)`. Також важливою частиною лабораторної роботи було набуття навичок з використання системи контролю версій файлів та спільної роботи `Git`.

На відміну від процедурної парадигми, де всі методи реалізовані в одному файлі вихідного коду, що викликає незручності в достатньо масштабних проектах, модульна парадигма базується на понятті «модуль», яке можна пояснити як самостійну програмну одиницю, що служить для виконання певної функції програми та для зв'язку з іншою частиною програми. В даній лабораторній роботі створений модуль було представлено у вигляді статичної бібліотеки (файл з розширенням `.a`), яка була підключена до проектів за допомогою лінкера та до файлу вихідного коду за допомогою новоствореного заголовкового файлу (файл з розширенням `.h`), що містить прототипи функцій, що містяться в статичній бібліотеці.

Статична бібліотека була створена шляхом відповідного проекту в застосунку `Code::Blocks`. Далі для подальшого інтегрування бібліотеки у проекти, написані мовою C++, файл `main.c` було замінено на `main.cpp`, після чого потребувалося додати новостворений файл у проект. У даному файлі було реалізовано функцію для вирахування виразу, заданого в умові завдання, що повертала значення результату. Після компіляції проекту статичної

бібліотеки в теці `\obj` було створено файл з розширенням `.a`, що і представляє з себе файл статичної бібліотеки.

Наступним кроком було створення заголовкового файлу, який містив прототип функції, реалізованої в статичній бібліотеці.

Далі було створено проект консольного додатка C++ під назвою `TestDriver`, Метою створення цього додатка була реалізація концепції модульного тестування. Модульне тестування, відоме як Unit-тестування, використовується для автоматизованого тестування модулів шляхом порівняння значення, що повертається функцією з модуля, яка приймає набір еталонних вхідних даних (аргументів), з еталонним результатом для відповідного набору. Важливою ідеєю юніт-тестування є така: якщо хоча б один з тестів провалився, після відлагодження модуля потрібно проводити повторне проведення тестів, що йшли перед провальним тестом, до моменту, коли усі тести проходять успішно.

Після цього було створено ПЗ з реалізацією функцій з наступного завдання. Важливою умовою було проектування функцій з інтерфейсом, який підходить для виводу у вихідний потік (`cout`). Для коректного виводу знаку © довелось намагатись писати через команди `string`, `cout`, `authorCopyright()` та просто вставляти знак копірайту але через невідомі навички та можливу помилку в програмі мені не вдалось зробити знак копірайту тож я вирішив просто зробити так (C).

Важливою частиною лабораторної роботи було використання системи контролю версій файлів та спільної роботи `Git`. Ідея системи контролю версій полягає у зручному зберіганні різних версій проекту в одному місці. Замість зберігання копій одного й того ж проекту, реєструються зміни в проекті, що дозволяє за потреби відкатити проект до потрібної версії (в даному контексті “версія” описує стан проект після певної послідовності змін). У випадку цієї лабораторної роботи проект зберігається на сайті `Github`. Для відсилення змін у проекті на сайт подрібно за допомогою застосунку `git-scm` клонувати репозиторій проекту на свій пристрій, зробити потрібні зміни. Далі додати потрібні файли в список для додання до комміту (структурна одиниця в архітектурі `git`), після чого потрібно сформулювати комміт з коментарем до нього. Останнім кроком є пуш(відсилення) комміту на сервер. Для сайту `Github` у застосунку `git-scm` потрібно попередньо авторизувати свій акаунт, який повинен мати доступ для змін до потрібного репозиторію. Історія змін складається з коммітів, кожен з яких має свій ідентифікатор, назву (коментар), дату додання та посилання на акаунт, від імені якого відбувся пуш. Система `git` дозволяє дізнаватись про те, які саме зміни були внесені, ким і коли, що значно спрощує командну розробку проектів.

ДОДАТОК А

Назва тестового набору Test Suite Description	TS_lab8_1
Назва проекту / Модуля Name of Project / Unit	ModulesChernoles.a
Рівень тестування Level of Testing	модульний / Unit Testing
Автор тест-сюїта Test Suite Author	Чернолес Кирило
Виконавець Implementer	Чернолес Кирило

Ід-р тест- кейса / Test Case ID	Вхідні дані / Input values	Очікуваний результат / Expected Result	Результат тестування / Test Result
TC_01	x = 0 z = 0	4	
TC_02	x = 1 z = 0	-39.2598	
TC_03	x = 0 z = 60	217863	
TC_04	x = 0 z = 1	9	
TC_05	x = 5 z = 0	4.49294	
TC_06	x = 9 z = 11	1482.26	
TC_07	x = 0 z = 2	23.3137	
TC_08	x = 450 z = 0	6.28607	
TC_09	x = 22 z = 64	264197	
TC_10	x = 11 z = 9	841.712	

ДОДАТОК Б

Лістинг вихідного коду проекту ModulesChernoles:

```
#include "main.h"
#include <cmath>
double s_calculation(double x,double z)
{ double S=(sqrt(1 + x) - 3 * cos(x))/(log(pow(x,2)) + 3 * sin(3.14 * x)) +
(pow((sqrt(pow(z,3)) + 2),2));
return S;
}
```

ДОДАТОК Б

Лістинг вихідного коду проекту TestDriver:

```
#include <iostream>
#include <main.h>
using namespace std;

int main()
{ setlocale(LC_ALL, "ukr");
  cout << "Чернолєс Кирило (C) | Академ-група:КБ-23 | ЦНТУ" << endl;
  int x,z;
  cout << "Введіть значення x та z" << endl;
  cin>> x;
  cin>> z;
  cout << s_calculation(x, z) << endl;
  return 0;
}
```


ДОДАТОК Г

Назва тестового набору Test Suite Description	TS_lab8_2
Назва проекту / ПЗ Name of Project / Software	Chernoless_task.exe
Рівень тестування Level of Testing	системний / System Testing
Автор тест-сюита Test Suite Author	Чернолес Кирило
Виконавець Implementer	Чернолес Кирило

Ід-р тест- кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування / Test Result
TC_01	<ol style="list-style-type: none"> Ввести "0 0" Натиснути Enter Ввести "6 300" Натиснути Enter 	<p>Чернолес Кирило (с) Академ-група:КБ-23 ЦНТУ</p> <p>Введіть значення a та b 0</p> <p>Введіть значення x та z В десятковій системі числення x та z відповідно: 6 300</p> <p>В шістнадцятковій системі числення x та z відповідно: 6 12с</p> <p>Результат виразу у функції s_calculation(): 27020788.543652</p>	passed
TC_02	<ol style="list-style-type: none"> Ввести "16 92" Натиснути Enter Ввести "72 15" Натиснути Enter 	<p>Чернолес Кирило (с) Академ-група:КБ-23 ЦНТУ</p> <p>Введіть значення a та b 1</p> <p>Введіть значення x та z В десятковій системі числення x та z відповідно: 72 15</p> <p>В шістнадцятковій системі числення x та z відповідно: 48 f</p> <p>Результат виразу у функції s_calculation(): 3612.773112</p>	passed
TC_03	<ol style="list-style-type: none"> Ввести "-83 79" Натиснути Enter Ввести "0 1204" Натиснути Enter 	<p>Чернолес Кирило (с) Академ-група:КБ-23 ЦНТУ</p> <p>Введіть значення a та b 1</p> <p>Введіть значення x та z В десятковій системі числення x та z відповідно: 0 1204</p> <p>В шістнадцятковій системі числення x та z відповідно: 0 4b4</p> <p>Результат виразу у функції s_calculation(): 1745504776.954350</p>	passed
TC_04	<ol style="list-style-type: none"> Ввести "-43 -5" Натиснути Enter Ввести "456 122" Натиснути Enter 	<p>Чернолес Кирило (с) Академ-група:КБ-23 ЦНТУ</p> <p>Введіть значення a та b 1</p> <p>Введіть значення x та z В десятковій системі числення x та z відповідно: 456 122</p> <p>В шістнадцятковій системі числення x та z відповідно: 1с8 7а</p> <p>Результат виразу у функції s_calculation(): 1821244.482228</p>	passed

TC_05	<ol style="list-style-type: none">1. Ввести "-567 123"2. Натиснути Enter3. Ввести "22 67890"4. Натиснути Enter	Чернолес Кирило (с) Академ-група:КБ-23 ЦНТУ Введіть значення a та b 1 Введіть значення x та z В десятичній системі числення x та z відповідно: 22 67890 В шістнадцятковій системі числення x та z відповідно: 16 10932 Результат виразу у функції s_calculation(): 312908617825890.125000	passed
-------	---	--	--------