МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 10

з навчальної дисципліни

“Базові методології та технології програмування”

РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ ОБРОБЛЕННЯ ДАНИХ СКЛАДОВИХ ТИПІВ З ФАЙЛОВИМ ВВЕДЕННЯМ/ВИВЕДЕННЯМ

ВИКОНАВ

студент академічної групи КН-24

Марина Середа

ПЕРЕВІРИЛА

викладачка кафедри кібербезпеки

та програмного забезпечення

Анастасія КОВАЛЕНКО

Кропивницький – 2025

**Мета роботи:** набуття ґрунтовних вмінь і практичних навичок реалізації у Code::Blocks IDE мовою програмування С++ програмних модулів створення й оброблення даних типів масив, структура, об’єднання, множина, перелік, перетворення типів даних, використання файлових потоків та функцій стандартних бібліотек для оброблення символьної інформації.

**Завдання до лабораторної роботи**

1. Реалізувати програмні модулі розв’язування задач 10.1–10.3 як складові статичної бібліотеки libModulesПрізвище.а (проект ModulesПрізвище лабораторних робіт №8–9).
2. Реалізувати тестовий драйвер автоматизованої перевірки програмних модулів розв’язування задач 10.1–10.3.

**Варіант 8**

Задача 10.1

Архітектура

1. Створення у модулі s\_calculate модулів name(), file(), processText(), outputFile(), developer () (реалізація завдання 10.1).
2. Створюємо консольної програми TestDriver і викликання новостворених функцій.
3. Результати програми додаються у вихідний файл.
4. Створюється набір TestSuite до модулів.

Вхідні дані:

string – fileContent

string - text

Вихідні дані:

string - virsh\_1 або

string virsh\_2

string – inf

string – text

Задача 10.2

Архітектура

1. Створення у модулі s\_calculate модулів ten\_two() і appendToFile\_ttt() (реалізація завдання 10.2).
2. Створюємо консольної програми TestDriver і викликання новостворених функцій.
3. Результати програми додаються у вихідний файл.
4. Створюється набір TestSuite до модулів.

Вхідні дані:

string fileContent

Вихідні дані:

string – текст

Задача 10.3

Архітектура

1. Створення у модулі s\_calculate модулів s\_calculation(), to\_string(), appendToFile() і printBinary() (реалізація завдання 10.3).
2. Створюємо консольної програми TestDriver і викликання новостворених функцій.
3. Результати програми додаються у вихідний файл.
4. Створюється набір TestSuite до модулів.

Вхідні дані:

double – x, y, z

int - b

Вихідні дані:

double - s

string - result

**Висновок:**

Під час виконання цієї лабораторної роботи я закріпив знання з модульного програмування та покращив свої навички розробки програм на мові C++ у середовищі Code::Blocks. Особливу увагу я приділила створенню та використанню програмних модулів, які розв’язують задачі 10.1–10.3, а також організації їх у вигляді статичної бібліотеки.

Я самостійно проектувала архітектуру програмного забезпечення, реалізовувала модулі, протестувала їх за допомогою спеціального драйвера та задокументувала результати. Я поглибила розуміння принципів модульного підходу та ефективного коду тестування. Виконання цієї роботи допомогло мені стати впевненішою в аналізі завдань і реалізації функціональних рішень.

1. Опанувала принцип розділення програми на незалежні модулі , що забезпечує масштабованість і зручність супровідного коду.
2. Навчилась створювати статичні бібліотеки , що дозволяють повторно використовувати функції в різних проектах без дублювання коду.
3. Застосувала заголовкові файли для структурування оголошень , що забезпечують зручність при роботі з великою кількістю функцій.
4. Окремо розробила кожен програмний модуль , який міг краще зрозуміти його логіку та тестувати незалежно.
5. Реалізувала архітектурний проект згідно зі структурою папок , яка відповідає практиці командної роботи над ПЗ.
6. Ознайомилася з підключенням зовнішніх модулів через#include , що зміцнило моє розуміння компіляції та зв'язування.
7. Навчилась оновлювати бібліотеку без зміни решти програми , що є основою для ефективної підтримки ПЗ.
8. Застосовувала принцип “одна функція — одне завдання” , який спрощує розуміння й тестування коду.
9. Освоїла створення окремого проекту для тестування функцій , що наближав мене до професійного підходу unit-тестування.
10. Побачила, як тестування окремих модулів покращує якість кінцевого продукту , даючи можливість виправити помилки на ранньому етапі.
11. Зрозуміла важливість ізоляції функцій для запобігання конфліктам між частинами коду.
12. Практично закріпила знання про зв'язок між файлами .cpp та .h.
13. Навчилась розпізнавати й вирішувати помилки компіляції, пов'язані з модулями.
14. Розробила проєктну документацію для кожного з модулів , що важливо для колективної розробки.
15. Оцінила ефективність логіки відокремлення обробки даних у функції , що значно спростило подальше тестування.
16. Навчилась використовувати змінні як аргументи між модулями , уникаючи глобальних змінних.
17. Застосувала підхід іменування модулів, що робить код читабельним.
18. Поглибила розуміння циклу компіляції й зв'язування проектів у C++.
19. Навчилася писати багаторазові функції, які можна використовувати в різних завданнях.
20. Узагальнила досвід побудови структурованих програм, які легко масштабувати.
21. Під час реалізації модулів я стикнулася із помилками, які допомогли мені краще зрозуміти логіку відладки.
22. Застосовувала реальний робочий процес командної розробки через Git.
23. Практикувалася у створенні README.md для документації проекту.
24. Зрозуміло, як важливо документувати функції інтерфейсу для полегшення командної роботи.
25. Опанувала підходи до написання функцій, які не залежать від глобального середовища — що забезпечує тестованість і стабільність.
26. Опанувала синтаксис роботи з ifstreamтаofstream , що дало можливість зчитувати та записувати текстову інформацію у файлі.
27. Розробила алгоритми зчитування даних із текстових файлів , що знадобляться в будь-якому реальному застосунку, який працює з файлами.
28. Використовувала потоки для запису результатів виконання модулів , що автоматизує аналіз результатів.
29. Ознайомилася з відкриттям файлів у різних режимах (читання, запис, дозапис) , що розширило можливості роботи з даними.
30. Практично вивчила функції eof(), open()таclose() , зрозуміла їхню роль у захисті від помилок при роботі з файлами.
31. Навчилася перевіряти, чи успішно відкрився файл перед зчитуванням , що передує аварійні завершення програми.
32. Реалізувала механізм знищення попереднього вмісту файлу за допомогою режимуios::trunc , що є частиним сценарієм у практиці.
33. Створила тестові файли вручну й автоматично , що допомогло зрозуміти помилку при підході роботи з вхідними даними.
34. Для тестування застосовувався принцип «один модуль — один файл» , що підвищило прозорість результатів.
35. Навчилась організовувати структуру папок (Software, TestSuite, Report) для зберігання тестів і звітів.
36. Зрозуміло, як читати текстовий файл пізніше і працювати з рядками у форматі C++.
37. Використовувала cstringдля обробки текстових даних, зокрема функції strlen, strcmp, strcat.
38. Написала тест-кейси з конкретними прикладами файлів, що формалізує процеси перевірки.
39. Використовувала константні масиви для формування вхідних даних та порівняння з очікуваним результатом.
40. Розробила функції, які приймають імена файлів як аргументи, що забезпечують гнучкість використання.
41. Поглибила знання про відмінності текстових і двох потоків.
42. Реалізувала обробку помилок відкриття файлів через перевірку is\_open().
43. Навчилася формувати автоматичні звіти про результати виконання функцій.
44. Дослідила механізми роботи з великими обсягами тексту у файлах.
45. Розробила алгоритм обробки тексту з можливістю подальшого аналізу.
46. Зрозуміла важливість стандарту ДСТУ 3008 у підготовці звіту.
47. Проаналізувала, як у реальних проектах ведеться реєстрація через текстові файли.
48. Вивчила способи захисту даних від випадкового перезапису.
49. Навчилася передавати результати тестів у вигляді логів, що зручно для перевірки та перевірки.
50. Поглибила розуміння обробки символьної інформації в текстовому вигляді, що є основою для розробки текстових інтерфейсів.

**Додатки**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_10.1 |
| Назва проекта / ПЗ  Name of Project / Software | ModulesSereda |
| Рівень тестування  Level of Testing | модульне / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Середа Марина |
| Виконавець  Implementer | Середа Марина |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| 1 | 1. Запуск програми. | Введіть ім'я файлу, який ви хочете відкрити: | passed |
| 2 | 1. Запуск програми. 2. Увести ttt.txt. | Введіть ім'я файлу, який ви хочете відкрити: ttt.txt  Результат записано у файл "result.txt".  ----------------------------  |Розробник: © Середа М. О. |  |Назва ЗВО: ЦНТУ |  |Місто: Кропивницький |  |Країна: Україна |  |Рік реалізації проєкту: 2025|  ----------------------------  ﻿?а?а?а?а?а  В хаті сонячній промінь косо  На долівку ляга з вікна...  Твої чорні шовкові коси  Припорошила вже сивина. | passed |
| 3 | 1. Запуск програми. 2. Увести ttt.txt. | Введіть ім'я файлу, який ви хочете відкрити: ttt.txt  Результат записано у файл "result.txt".  ----------------------------  |Розробник: © Середа М. О. |  |Назва ЗВО: ЦНТУ |  |Місто: Кропивницький |  |Країна: Україна |  |Рік реалізації проєкту: 2025|  ----------------------------  ﻿?а?а?а?а?аа  Коли малим ти перший став на ноги  Яка ж то радість матері була!  Від тихої колиски до порога  Вона тебе за руку провела.  94.659122  00011001 | passed |

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_10.2 |
| Назва проекта / ПЗ  Name of Project / Software | ModulesSereda |
| Рівень тестування  Level of Testing | модульне / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Середа Марина |
| Виконавець  Implementer | Середа Марина |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| 1 | 1. Запуск програми. | Введіть ім'я файлу, який ви хочете відкрити: | passed |
| 2 | 1. Запуск програми. 2. Увести ttt.txt. | Введіть ім'я файлу, який ви хочете відкрити: ttt.txt каталаУайа  Непарні літери: ктлУй  Час і дата: Thu Apr 17 10:00:21 2025 | passed |
| 3 | 1. Запуск програми. 2. Увести ttt. | Введіть ім'я файлу, який ви хочете відкрити: ttt  Помилка відкриття файлу "ttt". Перевірте, чи існує файл. | passed |

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_10.3 |
| Назва проекта / ПЗ  Name of Project / Software | ModulesSereda |
| Рівень тестування  Level of Testing | модульне / Modules Testing |
| Автор тест-сьюта  Test Suite Author | Середа Марина |
| Виконавець  Implementer | Середа Марина |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| 1 | 1. Запустити програму. | Введіть ім'я файлу, який ви хочете відкрити: | passed |
| 2 | 1. Запустити програму. 2. Увести . ttt.txt. 3. Увести 5. 4. Увести 3. 5. Увести 6. 6. Увести 23. | Введіть ім'я файлу, який ви хочете відкрити: ttt.txt  Введіть x: 5  Введіть y: 3  Введіть z: 6  Введіть b: 23  19.279636  00010111 | passed |
| 3 | 1. Запустити програму. 2. Увести ttt.txt. 3. Увести 0. 4. Увести 0. 5. Увести 0. 6. Увести 0. | Введіть ім'я файлу, який ви хочете відкрити: ttt.txt  Введіть x: 0  Введіть y: 0  Введіть z: 0  Введіть b: 0  0.000000  00000000 | passed |
| 4 | 1. Запустити програму. 2. Увести ttt.txt. 3. Увести 1. 4. Увести 1. 5. Увести 1. 6. Увести 1. | Введіть ім'я файлу, який ви хочете відкрити: ttt.txt  Введіть x: 1  Введіть y: 1  Введіть z: 1  Введіть b: 1  6.566981  00000001 | passed |
| 5 | 1. Запустити програму. 2. Увести ttt.txt. | Введіть ім'я файлу, який ви хочете відкрити: hjj.txt  Помилка відкриття файлу "hjj.txt". Перевірте, чи існує файл. | passed |

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_TestDriver |
| Назва проекта / ПЗ  Name of Project / Software | ModulesSereda |
| Рівень тестування  Level of Testing | модульне / Unit Testing |
| Автор тест-сьюта  Test Suite Author | Середа Марина |
| Виконавець  Implementer | Середа Марина |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| 1 | 1. Запуск програми. | Введіть ім'я файлу, який ви хочете відкрити: | passed |
| 2 | 1. Запуск програми. 2. Увести ttt.txt. | Введіть ім'я файлу, який ви хочете відкрити: ttt.txt  Результат записано у файл "result.txt".  ----------------------------  |Розробник: © Середа М. О. |  |Назва ЗВО: ЦНТУ |  |Місто: Кропивницький |  |Країна: Україна |  |Рік реалізації проєкту: 2025|  ----------------------------  ﻿?а?а?а?а?аа  Коли малим ти перший став на ноги  Яка ж то радість матері була!  Від тихої колиски до порога  Вона тебе за руку провела.  94.659122  00011001  Непарні літери: ктлУй  Час і дата: Thu Apr 17 10:00:21 2025 | passed |
| 3 | 1. Запуск програми. 2. Увести ttt. | Введіть ім'я файлу, який ви хочете відкрити: ttt  Помилка відкриття файлу "ttt". Перевірте, чи існує файл. | passed |

**Лістинги**

**ModulesSereda.cpp**

#include <iostream>

#include <cmath>

#include <cstdio>

#include <fstream>

#include <string>

#include <ctime>

using namespace std;

double s\_calculation(double x, double y, double z){

double s;

if(z != 0 && y != 0){

s = sqrt(abs(pow(z, 2) \* (1.0/2) \* y)) + ((M\_PI \* x + pow(M\_E, abs(y))) / y);

}

return s;

}

string name(){

string inputFilename;

cout << "Введіть ім'я файлу, який ви хочете відкрити: ";

cin >> inputFilename;

return inputFilename;

}

string file(string inputFilename){

ifstream inputFile(inputFilename);

string fileContent;

if (inputFile.is\_open()) {

string line;

while (getline(inputFile, line)) {

fileContent += line;

}

inputFile.close();

}

else {

cerr << "Помилка відкриття файлу \"" << inputFilename << "\". Перевірте, чи існує файл.\n";

}

return fileContent;

}

string processText(const string fileContent) {

int letterCount = 0;

for (char c : fileContent) {

if (isalpha(static\_cast<unsigned char>(c))) {

letterCount++;

}

}

string virsh\_1 = "\nВ хаті сонячній промінь косо\n"

"На долівку ляга з вікна...\n"

"Твої чорні шовкові коси\n"

"Припорошила вже сивина.";

string virsh\_2 = "\nКоли малим ти перший став на ноги\n"

"Яка ж то радість матері була!\n"

"Від тихої колиски до порога\n"

"Вона тебе за руку провела.";

if (letterCount % 2 == 0) {

return virsh\_1;

} else {

return virsh\_2;

}

}

void outputFile(string outputText\_1, string outputText\_2, string outputText\_3){

string outputFilename = "result.txt";

ofstream outputFile(outputFilename);

if (outputFile.is\_open()) {

outputFile << outputText\_1 << outputText\_2 << outputText\_3 << endl;

cout << "Результат записано у файл \"" << outputFilename << "\".\n";

outputFile.close();

} else {

cerr << "Помилка створення або відкриття файлу \"" << outputFilename << "\" для запису.\n";

}

}

string bykvu(string text){

for (char &c : text){

if (c == 'к' || c == 'т' || c == 'У' || c == 'л' || c == 'й'){

c = '?';

}

}

return text;

}

string developer (){

string inf = "\t\t\t\t\t\t ----------------------------\n"

"\t\t\t\t\t\t|Розробник: \u00A9 Середа М. О. |\n"

"\t\t\t\t\t\t|Назва ЗВО: ЦНТУ |\n"

"\t\t\t\t\t\t|Місто: Кропивницький |\n"

"\t\t\t\t\t\t|Країна: Україна |\n"

"\t\t\t\t\t\t|Рік реалізації проєкту: 2025|\n"

"\t\t\t\t\t\t ----------------------------\n";

return inf;

}

string ten\_two(string fileContent) {

string lettersOnly;

string result;

for (char c : fileContent) {

if (isalpha(static\_cast<unsigned char>(c))) {

lettersOnly += c;

}

}

for (size\_t i = 1; i < lettersOnly.length(); i += 2) { //lettersOnly.length() - довжина рядка/кільсть літер

result += lettersOnly[i];

}

time\_t now = time(0);

tm\* localtm = localtime(&now);

string timeString = asctime(localtm);

if (!timeString.empty() && timeString.back() == '\n') {

timeString.pop\_back();

}

return "\nНепарні літери: " + result + "\nЧас і дата: " + timeString;

}

string printBinary(int b) {

string result;

for (int i = 7; i >= 0; --i) {

result += ((b >> i) & 1) ? '1' : '0';

}

return result;

}

void appendToFile(const string text) {

ofstream file("result.txt", ios::app);

if (file.is\_open()) {

file << text << endl;

file.close();

} else {

cerr << "Не вдалося відкрити файл для запису!" << endl;

}

}

void appendToFile\_ttt(const string text, string file\_1) {

ofstream file(file\_1, ios::app);

if (file.is\_open()) {

file << text << endl;

file.close();

} else {

cerr << "Не вдалося відкрити файл для запису!" << endl;

}

}

**ModulesSereda.h**

#ifndef MODULESSEREDA\_H\_INCLUDED

#define MODULESSEREDA\_H\_INCLUDED

using namespace std;

double s\_calculation(double x, double y, double z);

string name();

string file(string inputFilename);

string processText(const string fileContent);

void outputFile(string outputText\_1, string outputText\_2, string outputText\_3);

string bykvu(string text);

string developer ();

string ten\_two(string fileContent);

string printBinary(int b);

void appendToFile(const string text);

void appendToFile\_ttt(const string text, string file\_1);

#endif // MODULESSEREDA\_H\_INCLUDED

**TetsDriver.cpp**

#include <iostream>

#include <cmath>

#include <cstdio>

#include <fstream>

#include <string>

#include <ctime>

#include "ModulesSereda.h"

using namespace std;

int main(){

string name\_1 = name();

string fileContent = file(name\_1);

string inf = developer ();

string td = bykvu(fileContent);

string pt = processText(fileContent);

outputFile(inf, td, pt);

string tw = ten\_two(fileContent);

appendToFile\_ttt(tw, name\_1);

double x = 0, y = 0, z = 0;

cout << "Введіть x: ";

cin >> x;

cout << "Введіть y: ";

cin >> y;

cout << "Введіть z: ";

cin >> z;

double s = s\_calculation(x, y, z);

string result = to\_string(s);

appendToFile(result);

int b = 0;

cout << "Введіть b: ";

cin >> b;

string ba = printBinary(b);

appendToFile(ba);

return 0;

}