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 \_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_ Вероніка СИТЕНКОВА

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ганна ДРЄЄВА

Кропивницький – 2024

**Тема:** Реалізація програмних модулів оброблення даних складових типів з файловим введенням/виведенням

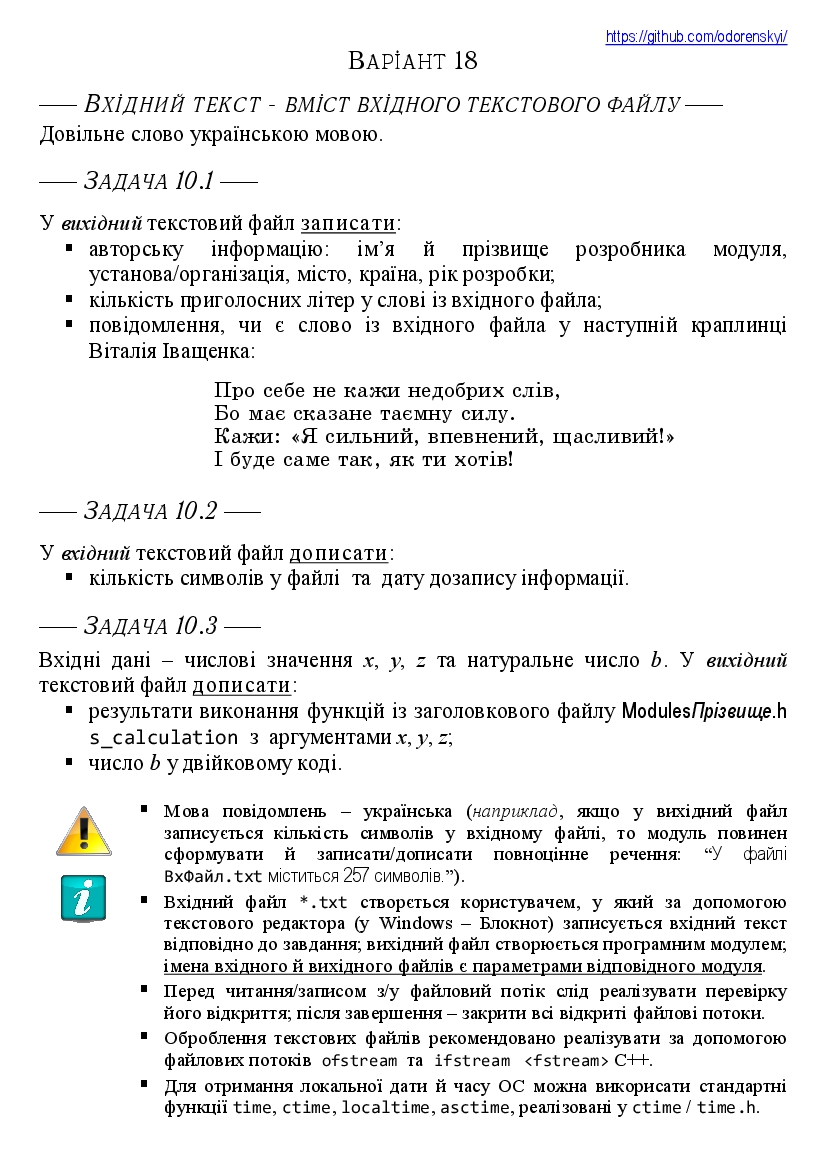
**Мета**: Набуття ґрунтовних вмінь і практичних навичок реалізації у Code::Blocks IDE мовою програмування С++ програмних модулів створення й оброблення даних типів масив, структура, об’єднання, множина, перелік, перетворення типів даних, використання файлових потоків та функцій стандартних бібліотек 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 18**

****

**Задача 10.1**

Вхідні дані: авторська інформація, ua\_word.txt

Вихідні дані: consonant, повідомлення, чи є слово

Початок

1. Записати у вихідний файл авторську інформацію

2. Порахувати кількість приголосних літер в слові з вихідного файлу.

3. Записати у вихідний файл кількість приголосних літер.

4. Перевірити, чи є слово з вхідного файлу в краплинці.

5. Записати повідомлення щодо 4 пункту

Кінець

**Задача 10.2**

Вхідні дані: ua\_word.txt

Вихідні дані: symbol, дата дозапису

Початок

1. Порахувати кількість символів у вхідному файлі.

2. Дозаписати 1 пункт у вхідний файл

3. Дозаписати дату дозапису.

Кінець

**Задача 10.3**

Вхідні дані: x, y, z, b – int;

Вихідні дані: S, B - int

Початок

1. Дописати у вихідний файл результат s\_calculation

2. Дописати b у двійковому коді за допомогою функції to\_binary з бібліотеки ModulesSytenkova.h

Кінець

:====== TS\_10\_1 ======:

TC\_01

Вхідні дані:

input\_file: input\_1.txt

Вміст файлу відповідає тест-кейсу? (y/n):

Результат: passed

TC\_02

Вхідні дані:

input\_file: input\_2.txt

Вміст файлу відповідає тест-кейсу? (y/n):

Результат: passed

TC\_03

Вхідні дані:

input\_file: input\_3.txt

Вміст файлу відповідає тест-кейсу? (y/n):

Результат: passed

:====== TS\_10\_2 ======:

TC\_01

Вхідні дані:

input\_file: input\_4.txt

Інформацію успішно дописано у файл: input\_4.txt

Вміст файлу відповідає тест-кейсу? (y/n):

Результат: passed

TC\_02

Вхідні дані:

input\_file: input\_5.txt

Інформацію успішно дописано у файл: input\_5.txt

Вміст файлу відповідає тест-кейсу? (y/n):

Результат: passed

TC\_03

Вхідні дані:

input\_file: input\_6.txt

Інформацію успішно дописано у файл: input\_6.txt

Вміст файлу відповідає тест-кейсу? (y/n):

Результат: passed

:====== TS\_10\_3 ======:

TC\_01

Вхідні дані:

x: 1

z: 1

b: 10

output\_file: output\_4.txt

Вміст файлу відповідає тест-кейсу? (y/n):

Результат: passed

TC\_02

Вхідні дані:

x: 2

z: 2

b: 182

output\_file: output\_5.txt

Вміст файлу відповідає тест-кейсу? (y/n):

Результат: passed

TC\_03

Вхідні дані:

x: -1

z: 0

b: 255

output\_file: input\_6.txt

Вміст файлу відповідає тест-кейсу? (y/n):

Результат: passed

**Лістинг ModulesSytenkova**

**#**include <iostream>

#include <math.h>

#include <fstream>

#include <string>

#include <ctime>

using namespace std;

double s\_calculation (double x, double z)

{

double S = ((0.5)\*(pow(((abs(2\*z - pow(x, 2)))/ sin(x)), 3)))/(pow(1+(abs(cos(x))), 0.5) + 2 \* 3.1415);

return S;

}

float sale (float buy)

{

if (buy<=0)

{

std::cout << "Вартість має бути додатньою.";

return 0;

}

else

{

if (buy<1000) buy = buy\*0.99;

else if (buy<2000) buy = buy\*0.95;

else if (buy<3000) buy = buy\*0.93;

else if (buy<5000) buy = buy\*0.9;

else if (buy<1000) buy = buy\*0.85;

else buy = buy\*0.8;

return buy;

}

}

void whiteness (int siz)

{

switch (siz)

{

case 44:

std::cout << "Відповідник у системі Франції - 2\nВідповідник у системі Великобританії - 34\nВідповідник у міжнародній системі - S";

break;

case 46:

std::cout << "Відповідник у системі Франції - 3\nВідповідник у системі Великобританії - 36\nВідповідник у міжнародній системі - M";

break;

case 48:

std::cout << "Відповідник у системі Франції - 4\nВідповідник у системі Великобританії - 38\nВідповідник у міжнародній системі - L";

break;

case 50:

std::cout << "Відповідник у системі Франції - 5\nВідповідник у системі Великобританії - 40\nВідповідник у міжнародній системі - XL";

break;

case 52:

std::cout << "Відповідник у системі Франції - 6\nВідповідник у системі Великобританії - 42\nВідповідник у міжнародній системі - XXL";

break;

case 54:

std::cout << "Відповідник у системі Франції - відсутній\nВідповідник у системі Великобританії - 44\nВідповідник у міжнародній системі - XXXL";

break;

default:

std::cout << "Такого розміру не існує.";

}

}

int to\_binary (int N)

{

int times0 = 0;

int times1 = 0;

if (N < 0 || N > 80000) return 0;

else

{

int bites = sizeof(N)\*8;

for (int i = 0; i < bites; ++i)

{

if (((N >> i) & 1) == 0) times0++;

else times1++;

}

return ((((N >> 5) & 1) == 1) ? times1 : times0);

}

}

void task\_ten\_one (string input\_file, string output\_file)

{

ifstream inFile(input\_file);

ofstream outFile(output\_file);

if (!inFile || !outFile)

{

cout << "Помилка відкриття файлу" << endl;

return;

}

string word;

inFile >> word;

outFile << ":----------------------------------------------------------:" << endl;

outFile << " Cитенкова Вероніка" << endl;

outFile << " Центральноукраїнський національний технічний університет" << endl;

outFile << " м. Кропивницький, Україна" << endl;

outFile << " 2024" << endl;

outFile << ":------------------ © All rights reserved -----------------:" << endl;

outFile << endl;

string consonants = "бвгґджзйклмнпрстфхцчшщ";

int consonantCount = 0;

for (char c : word)

{

char lower\_c = tolower(c);

for (char consonant : consonants)

{

if (lower\_c == consonant)

{

consonantCount++;

break;

}

}

}

outFile << "Кількість приголосних літер у вхідному файлі: " << consonantCount << endl;

string droplet = "Про себе не кажи недобрих слів, "

"Бо має сказане таємну силу. "

"Кажи: «Я сильний, впевнений, щасливий!» "

"І буде саме так, як ти хотів!";

for (char &c : droplet)

{

c = tolower(c);

}

for (char &c : word)

{

c = tolower(c);

}

bool found = false;

for (size\_t i = 0; i <= droplet.length() - word.length(); ++i)

{

size\_t j = 0;

while (j < word.length() && droplet[i + j] == word[j])

{

++j;

}

if (j == word.length())

{

found = true;

break;

}

}

if (found)

{

outFile << "Слово \"" << word << "\" знайдено у краплинці Віталія Іващенка." << endl;

}

else

{

outFile << "Слово \"" << word << "\" не знайдено у краплинці Віталія Іващенка." << endl;

}

inFile.close();

outFile.close();

}

void task\_ten\_two(const string& input\_file)

{

ifstream inputFile(input\_file);

if (!inputFile)

{

cout << "Не вдалося відкрити файл" << endl;

return;

}

time\_t time\_now;

time\_now = time(NULL);

inputFile.seekg(0, ios::end);

streampos fileSize = inputFile.tellg();

inputFile.close();

ofstream outputFile(input\_file, ios::app);

if (!outputFile)

{

cout << "Не вдалося відкрити файл для дозапису" << endl;

return;

}

outputFile << "\nКількість символів у вхідному вайлі: " << fileSize << endl;

outputFile << "Дата дозапису: " << ctime(&time\_now) << endl;

outputFile.close();

cout << "Інформацію успішно дописано у файл: " << input\_file << endl;

}

void task\_ten\_three(int x, int z, int b, const string& output\_file)

{

ofstream outputFile(output\_file, ios::app);

float S = s\_calculation(x, z);

outputFile << "\nРезультат функції s\_calculation: " << S << endl;

string binary = "";

while (b > 0)

{

int remainder = b % 2;

binary = std::to\_string(remainder) + binary;

b /= 2;

}

outputFile << "\nb у двійковому коді: " << binary << endl;

}

**Лістинг TestDriver**

#include <iostream>

#include <fstream>

#include <conio.h>

#include <string>

#include "ModulesSytenkova.h"

using namespace std;

#define TEST\_CASES 3

void TestDriver\_first\_task(int test\_case, string input\_file, string output\_file)

{

cout << "TC\_0" << test\_case << endl;

cout << "Вхідні дані:\ninput\_file: " << input\_file << endl;

task\_ten\_one(input\_file, output\_file);

cout << "Вміст файлу відповідає тест-кейсу? (y/n): " ;

char check;

cin >> check;

if (check == 'y')

{

cout << endl << "Результат: passed" << endl;

}

else

{

cout << endl << "Результат: failed" << endl;

}

cout << endl;

}

void TestDriver\_second\_task(int test\_case, string input\_file)

{

cout << "TC\_0" << test\_case << endl;

cout << "Вхідні дані:\ninput\_file: " << input\_file << endl;

task\_ten\_two(input\_file);

cout << "Вміст файлу відповідає тест-кейсу? (y/n): " ;

char check;

cin >> check;

if (check == 'y')

{

cout << endl << "Результат: passed" << endl;

}

else

{

cout << endl << "Результат: failed" << endl;

}

cout << endl;;

}

void TestDriver\_third\_task(int test\_case, int x, int z, int b, string output\_file)

{

cout << "TC\_0" << test\_case << endl;

cout << "Вхідні дані:\nx: " << x << "\nz: " << z << "\nb: " << b << endl;

cout << "output\_file: " << output\_file << endl;

task\_ten\_three(x, z, b, output\_file);

cout << "Вміст файлу відповідає тест-кейсу? (y/n): " ;

char check;

cin >> check;

if (check == 'y')

{

cout << endl << "Результат: passed" << endl;

}

else

{

cout << endl << "Результат: failed" << endl;

}

cout << endl;;

}

int main()

{

system("chcp 1251 & cls");

cout << ":====== TS\_10\_1 ======:" << endl;

const string first\_task[TEST\_CASES][2] = {

{"input\_1.txt", "output\_1.txt"},

{"input\_2.txt", "output\_2.txt"},

{"input\_3.txt", "output\_3.txt"}

};

for (int i = 0; i < TEST\_CASES; i++)

{

TestDriver\_first\_task(i + 1, first\_task[i][0], first\_task[i][1]);

}

cout << ":====== TS\_10\_2 ======:" << endl;

const string second\_task[TEST\_CASES] = {

"input\_4.txt",

"input\_5.txt",

"input\_6.txt"

};

for (int i = 0; i < TEST\_CASES; i++)

{

TestDriver\_second\_task(i + 1, second\_task[i]);

}

cout << ":====== TS\_10\_3 ======:" << endl;

const string third\_task[TEST\_CASES][4] = {

{"1", "1", "10", "output\_4.txt"},

{"2", "2", "182", "output\_5.txt"},

{"-1", "0", "255", "output\_6.txt"}

};

for (int i = 0; i < TEST\_CASES; i++)

{

TestDriver\_third\_task(i + 1, stoi(third\_task[i][0]), stoi(third\_task[i][1]), stoi(third\_task[i][2]), third\_task[i][3]);

}

system("pause");

return 0;

}

Висновок:

Для виконання лабораторної роботи, я користувалась методичними вказівками. Вони значно полегшують роботу. Із-за цього план моїх дій, описаних далі у висновку, відповідають порядку в методичних вказівках.

Коли перейшла до виконання цієї лабораторної роботи, я ознайомилась із поставленими задачами. Провела аналіз вхідних та вихідних даних, склала алгоритми для кожної задачі.

Наступним кроком розробила тест-сьюти для задач. Спочатку помилково розробила їх у звіті. Потім перенесла їх в окремі файли у папці TestSuite.

Далі почала безпосереднє виконання завдання – реалізації поставлених задач. Це зайняло не так багато часу, задачі доволі прості.

Перейшовши до тестування виникли певні проблеми. Зупинилась на напівавтоматизованому тестуванні, оскільки не до кінця розуміла як реалізувати автоматизоване. Під час спроб була створена наступна функція:

void Preliminary\_Steps()

{

ofstream infile\_1("input\_1.txt");

ofstream infile\_2("input\_2.txt");

ofstream infile\_3("input\_3.txt");

ofstream infile\_4("input\_4.txt");

ofstream infile\_5("input\_5.txt");

ofstream infile\_6("input\_6.txt");

infile\_1 << "сильний";

infile\_2 << "хотів";

infile\_3 << "піон";

infile\_4 << "полуниця";

infile\_5 << "блокнот";

infile\_6 << "краплинка";

}

Звичайно, її можна було використовувати, але великої користі вона не мала. Можливо, це із-за невеликого досвіду роботи з файлами, але варіант власноруч налаштовувати файли виглядав надійніше. Звичайно, за умови, що кількість тест-кейсів була б великою, то це мало б сенс, але кожну функцію можна перевірити за відносно невелику кількість тест-кейсів.

Також для реалізації тестового драйвера обрала використання 3 функцій, відповідно до кількості завдань лабораторної для зручності.

За виконання цієї лабораторної роботи я отримала навички роботи з файлами та закріпила знання, отримані на лекції. Але хочу відмітити, що в цій лабораторній тестування використовується не стільки для самого тестування, скільки для набуття вмінь тестування.