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 КБ-23

\_\_\_\_\_\_\_\_ Олексій КОНСТАНТИНОВ

ПЕРЕВІРИЛА

викладач кафедри кібербезпеки

та програмного забезпечення

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

Кропивницький – 2023

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

**МЕТА:** Набути ґрунтовних вмінь і практичних навичок 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:**

1. Реалізувати програмні модулі розв’язування задач 10.1–10.3 як складові статичної бібліотеки libModulesKonstantynov.h (проєкт ModulesKonstantynov лабораторних робіт №8–9).

2. Реалізувати тестовий драйвер автоматизованої перевірки програмних модулів розв’язування задач 10.1–10.3.

**Варіант 9**

Задача 10.1.

У вихідний текстовий файл записати авторську інформацію (ім’я, прізвище, організація, місто, країна, рік розробки), текст із вхідного файлу з переставленими місцями абзацами (перший з останнім, другий с передостаннім і тд) й повідомлення до української чи англійської мови належить текст

1) Аналіз задачі 10.1:

Вхідні дані: текстовий файл з довільним текстом

Вихідні дані: вхідний файл з авторською інформацією, переставленими абзацами та повідомленням про мову

2) Алгоритм:

1. Початок програми.

2. Відкриття файлу за вказаним шляхом.

3. Перевірка, чи відкриття файлу було успішним. Якщо ні, виведення повідомлення про помилку.

4. Якщо файл вдалося відкрити, створення списку для зберігання тексту.

5. Зчитування кожного рядка тексту з файлу та додавання його до списку.

6. Реверсування порядку рядків у списку (тобто з першого до останнього).

7. Перевірка кожного символу в файлі для визначення мови тексту:

- Якщо символ знаходиться в діапазоні українських символів, то вважається, що текст українською мовою.

- Якщо символ належить англійському алфавіту, то вважається, що текст англійською мовою.

- Якщо символ не входить ні в один із цих діапазонів, аналіз продовжується.

8. Додавання інформації про автора до файлу.

9. Запис кожного рядка тексту зі списку у файл.

10. Запис визначеної мови тексту у файл.

11. Закриття файлу.

3) Лістинг модуля й тестового драйвера знаходиться у додатках 1 та 2 відповідно.

Задача 10.2. У вхідний текст дописати кількість ком та крапок у ньому, дату та час дозапису.

1) Аналіз задачі 10.2:

Вхідні дані: текстовий файл з довільним текстом

Вихідні дані: вхідний файл з кількістю ком та крапок і датою й часом дозапису

2)Алгоритм:

1. Початок програми.

2. Відкриття файлу за вказаним шляхом з можливістю читання та запису, а також дозапису в кінець файлу.

3. Перевірка, чи відкриття файлу було успішним. Якщо ні, виведення повідомлення про помилку.

4. Якщо файл вдалося відкрити, ініціалізація лічильника ком і крапок.

5. Зчитування кожного символу з файлу та перевірка, чи це кома або крапка. Якщо так, збільшення лічильника.

6. Отримання поточної дати та часу.

7. Запис у файл інформації про кількість ком і крапок, а також поточну дату та час.

8. Закриття файлу.

3) Лістинг модуля й тестового драйвера знаходиться у додатках 1 та 2 відповідно.

Задача 10.3.

У вихідний файл дописати результати виконання функції s\_calculation із заголовкового файлу ModulesKonstantynov з аргументами x, y, z й число b у двійковому коді.

1) Аналіз задачі 10.3:

Вхідні дані: текстовий файл з довільним текстом, z, x, y, b.

Вихідні дані: вхідний файл з дописаними результатом виконання функції й числом у двійковому коді.

2) Алгоритм:

1. Початок програми.

2. Відкриття файлу за вказаним шляхом з можливістю читання та запису, а також дозапису в кінець файлу.

3. Перевірка, чи відкриття файлу було успішним. Якщо ні, виведення повідомлення про помилку.

4. Якщо файл вдалося відкрити, обчислення виразу s\_calculation(x, y, z) і запис результату у файл.

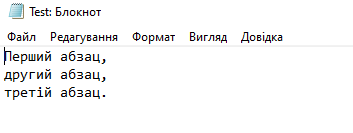
5. Перетворення числа b з десяткової системи числення у двійкову та запис цієї інформації у файл.

6. Закриття файлу.

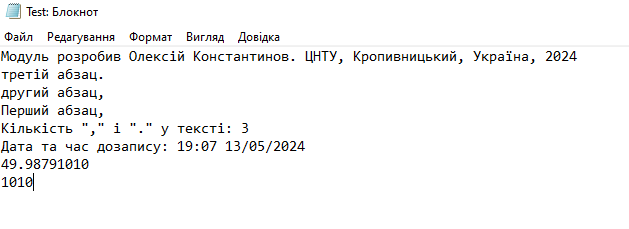
3) Лістинг модуля й тестового драйвера знаходиться у додатках 1 та 2 відповідно.

Результат виконання модульного тестування:

Вхідний файл:



Вихідний файл:



Висновок. У ході виконання лабораторної роботи, я опрацював роботу з файлами, доповнив модуль новими функціями, які здатні додавати нову інформацію у файл, міняти місцями текст у файлі, рахувати кількість елементів у файлі, вказувати дату та час внесення змін, та додавати інформацію, отриману з інших функцій.

Додаток 1. Вихідний код TestDriver

#include <string>;

#include "../../../../lab08/prj/ModulesKonstantynov/ModulesKonstantynov.h"

int main() {

std::string testf = "Test.txt";

task1(testf);

task2(testf);

task3(testf,10,20,30,10);

return 0;

}

Додаток 2. Вихідний код ModulesKonstantynov

#define \_USE\_MATH\_DEFINES

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <cmath>

#include <bitset>

#include <string>

#include <fstream>

#include <iostream>

#include <Windows.h>

#include <list>

using namespace std;

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

float s = log(fabs(sin(z))) + (0.5 \* x \* x - pow(fabs((pow((y + z), 2)) - pow(x, 5)), 0.5) / (10 \* M\_PI \* pow(z, 4)));

return s;

}

int getGrade(float waveHeight) {

if (waveHeight == 0) {

return 0;

}

else if (waveHeight > 0 && waveHeight <= 0.1) {

return 1;

}

else if (waveHeight > 0.1 && waveHeight <= 0.5) {

return 2;

}

else if (waveHeight > 0.5 && waveHeight <= 1.25) {

return 3;

}

else if (waveHeight > 1.25 && waveHeight <= 2.5) {

return 4;

}

else if (waveHeight > 2.5 && waveHeight <= 4) {

return 5;

}

else if (waveHeight > 4 && waveHeight <= 6) {

return 6;

}

else if (waveHeight > 6 && waveHeight <= 9) {

return 7;

}

else if (waveHeight > 9 && waveHeight <= 14) {

return 8;

}

else if (waveHeight > 14) {

return 9;

}

else return 10;

}

int temperature(int tFar) {

int tCel = (tFar - 32) \* 5 / 9;

return tCel;

}

int primalNum(int number) {

if (number <= 1 || number > 65535) {

return -1;

}

for (int i = 2; i < number; ++i) {

if (number % i == 0) {

return -1;

}

else {

return (number & 1) ? 16 - std::bitset<16>(number).count() : std::bitset<16>(number).count();

}

}

}

void task1(string file) {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char c;

fstream fs;

fs.open(file, fstream::in | fstream::out);

if (!fs.is\_open()) {

cout << "Помилка відкриття файлу";

}

else {

list<string> text;

string paragraph, language;

while (getline(fs, paragraph)) {

text.push\_back(paragraph);

}

text.reverse();

fs.seekg(0);

while (fs.get(c)) {

if ((c >= (char)0xC0 && c <= (char)0xDF) || (c >= (char)0xE0 && c <= (char)0xFF)) {

language = "Текст у файлі українською мовою\n";

break;

}

else if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z')) {

language = "Текст у файлі англійською мовою\n";

break;

}

}

fs << "Модуль розробив Олексій Константинов. ЦНТУ, Кропивницький, Україна, 2024\n";

for (const auto& str : text) {

fs << str << endl;

}

fs << language;

fs.close();

}

}

void task2(string file) {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

fstream fs;

int counter = 0;

char c;

fs.open(file, fstream::in | fstream::out | fstream::app);

if (!fs.is\_open()) {

cout << "Помилка відкриття файлу";

}

else {

while (fs.get(c)) {

if (c == '.' || c == ',') {

counter++;

}

}

time\_t current\_time = time(nullptr);

tm\* local\_time = localtime(&current\_time);

fs << "Кількість \",\" і \".\" у тексті:" << counter << endl << "Дата та час дозапису: " << asctime(local\_time) << endl;

fs.close();

}

}

void task3(string file, int x, int y, int z, int b) {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

string bin;

fstream fs;

fs.open(file, fstream::in | fstream::out | fstream::app);

if (!fs.is\_open()) {

cout << "Помилка відкриття файлу";

}

else {

fs << s\_calculation(x, y, z);

while (b > 0) {

bin = to\_string(b % 2) + bin;

b /= 2;

}

fs << bin;

fs.close();

}

}