

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 файлами введенням/виведенням**

ВИКОНАВ

студент академічної групи КБ-23

Сергій Козирь

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

Ганна ДРЄЄВА

Мета роботи: полягає у набутті ґрунтовних вмінь і практичних навичок реалізації у Code::Blocks IDE мовою програмування C++ 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 символічної інформації.

Час виконання лабораторної роботи

4 академічні години.

Обладнання, матеріали, програмні засоби

Для виконання лабораторної роботи необхідні: – персональний комп'ютер з ОС Windows XP / Vista / 7 / 8.x / 10, Linux 32-bit / 64-bit або Mac OS X; – вільне кросплатформове середовище розроблення програмного забезпечення Code::Blocks (www.codeblocks.org) для платформи Windows XP / Vista / 7 / 8.x / 10, Linux 32-bit / 64-bit, або Mac OS X; – текстовий редактор (OpenOffice Writer, Microsoft Word або ін.); – Git-репозиторій <https://github.com/odorenskyi/student-name>.

Завдання до лабораторної роботи

ВАРІАНТ 17

— ВХІДНИЙ ТЕКСТ - ВМІСТ ВХІДНОГО ТЕКСТОВОГО ФАЙЛУ —

Довільний текст українською мовою.

— ЗАДАЧА 10.1 —

У вихідний текстовий файл записати:

- авторську інформацію: ім'я й прізвище розробника модуля, установа/організація, місто, країна, рік розробки;
- кількість абзаців у тексті із вхідного файлу;
- повідомлення, чи є у тексті з вхідного файлу слова “Україна”, “університет” та “блокнот” (у програмі слід реалізувати розрізнення слів, наприклад, “університет” та “університету” тощо).

— ЗАДАЧА 10.2 —

У вихідний текстовий файл дописати:

- кількість крапок у ньому, дату й час дозапису інформації.

— ЗАДАЧА 10.3 —

Вхідні дані – числові значення x , y , z та натуральне число b . У вихідний текстовий файл дописати:

- результати виконання функцій із заголовкового файлу `Modules/Прізвище.h` `s_calculation` з аргументами x , y , z ;
- число b у двійковому коді.



- Мова повідомлень – українська (наприклад, якщо у вихідний файл записується кількість символів у вхідному файлі, то модуль повинен сформувати й записати/дописати повноцінне речення: “У файлі `ВхФайл.txt` міститься 257 символів.”).
- Вхідний файл `*.txt` створюється користувачем, у який за допомогою текстового редактора (у Windows – Блокнот) записується вхідний текст відповідно до завдання; вихідний файл створюється програмним модулем; імена вхідного й вихідного файлів є параметрами відповідного модуля.
- Перед читання/записом з/у файловий потік слід реалізувати перевірку його відкриття; після завершення – закрити всі відкриті файлові потоки.
- Оброблення текстових файлів рекомендовано реалізувати за допомогою файлових потоків `ofstream` та `ifstream` <fstream> C++.
- Для отримання локальної дати й часу ОС можна використати стандартні функції `time`, `ctime`, `localtime`, `asctime`, реалізовані у `ctime / time.h`.

Лістинг 10.1

```
#include <iostream>
```

```
#include <fstream>
#include <sstream>
#include <string>
#include <vector>
#include <algorithm>
```

```
bool containsWord(const std::string &text, const std::string &word) {
    std::string delimiters = " ,!?:;\n\t";
    size_t pos = text.find(word);
    while (pos != std::string::npos) {
        if ((pos == 0 || delimiters.find(text[pos - 1]) != std::string::npos) &&
            (pos + word.length() == text.length() || delimiters.find(text[pos +
word.length()]) != std::string::npos)) {
            return true;
        }
        pos = text.find(word, pos + 1);
    }
    return false;
}
```

```
void searchWords(const std::string &text, bool &hasUkraine, bool
&hasUniversity, bool &hasNotepad) {
    hasUkraine = containsWord(text, "Україна");
    hasUniversity = containsWord(text, "університет") || containsWord(text,
"університету");
    hasNotepad = containsWord(text, "блокнот");
}
```

```
int main() {
    std::ifstream inputFile("input.txt");
```

```
std::ofstream outputFile("output.txt");

if (!inputFile.is_open()) {
    std::cerr << "Не вдалося відкрити вхідний файл!" << std::endl;
    return 1;
}

// Авторська інформація
std::string authorName = "Сергій Козирь";
std::string organization = "Центральноукраїнський національний  
технічний університет";
std::string city = "Кропивницький";
std::string country = "Україна";
std::string year = "2024";

// Зчитування вхідного файлу
std::stringstream buffer;
buffer << inputFile.rdbuf();
std::string text = buffer.str();

// Кількість абзаців
size_t paragraphCount = std::count(text.begin(), text.end(), '\n') + 1;

// Перевірка наявності слів
bool hasUkraine = false;
bool hasUniversity = false;
bool hasNotepad = false;
searchWords(text, hasUkraine, hasUniversity, hasNotepad);

// Запис результатів у вихідний файл
```

```

        outputFile << "Авторська інформація: " << authorName << ", " <<
organization << ", " << city << ", " << country << ", " << year << std::endl;
        outputFile << "Кількість абзаців: " << paragraphCount << std::endl;
        outputFile << "Наявність слова 'Україна': " << (hasUkraine ? "Так" :
"Hi") << std::endl;
        outputFile << "Наявність слова 'університет': " << (hasUniversity ? "Так"
: "Hi") << std::endl;
        outputFile << "Наявність слова 'блокнот': " << (hasNotepad ? "Так" :
"Hi") << std::endl;
        outputFile << "Модуль було розроблено студентом
Центральноукраїнського національного технічного університету Козирь
Сергій, м. Кропивницький. Україна. 2024 рік." << std::endl;

```

```

        inputFile.close();
        outputFile.close();

```

```

        return 0;

```

```

    }

```

Лістинг 10.2

```

#include <iostream>
#include <fstream>
#include <ctime>
#include <bitset>

```

```

int countDots(const std::string& text) {
    int count = 0;
    for (char c : text) {
        if (c == '.') {
            count++;
        }
    }
}

```

```

    }
    return count;
}

std::string getCurrentDateTime() {
    std::time_t now = std::time(nullptr);
    char buf[80];
    std::strftime(buf, sizeof(buf), "%Y-%m-%d %H:%M:%S",
std::localtime(&now));
    return std::string(buf);
}

std::string toBinary(int number) {
    return std::bitset<32>(number).to_string(); // 32-бітове представлення
}

int main() {
    setlocale (LC_ALL, "ukr");
    std::string inputFileName = "input.txt";
    std::ifstream inputFile(inputFileName);
    if (!inputFile.is_open()) {
        std::cerr << "Не вдалося відкрити файл для читання." << std::endl;
        return 1;
    }

    std::string content((std::istreambuf_iterator<char>(inputFile)),
std::istreambuf_iterator<char>());
    inputFile.close();

    int dotCount = countDots(content);

```

```
std::string dateTime = getCurrentDateTime();  
int number = 123; // Замість 123 можна вставити будь-яке число  
std::string binaryNumber = toBinary(number);
```

```
std::ofstream outputFile(inputFileName, std::ios_base::app); //
```

Відкриваємо файл для дозапису

```
if (!outputFile.is_open()) {  
    std::cerr << "Не вдалося відкрити файл для запису." << std::endl;  
    return 1;  
}  
  
outputFile << "\nКількість крапок: " << dotCount << std::endl;  
outputFile << "Дата і час дозапису: " << dateTime << std::endl;  
outputFile << "Число у двійковому коді: " << binaryNumber << std::endl;  
  
outputFile.close();  
  
return 0;  
}
```

Лістинг 10.3

```
#include <iostream>  
#include <fstream>  
#include <bitset>  
#include <cmath>
```

```
// Оголошення функції з "Modules.h"
```

```
double s_calculation(double x, double y, int z) {  
    // Приклад реалізації, яка обчислює якусь функцію від x, y і z  
    return x * y + std::pow(z, 2);  
}
```



```
std::string toBinary(int number) {  
    return std::bitset<32>(number).to_string(); // 32-бітове представлення  
}
```

```
int main() {  
    std::string inputFileName = "input.txt";  
    std::string outputFileName = "output.txt";
```

```
    double x, y;  
    int z;
```

```
    std::ifstream inputFile(inputFileName);  
    if (!inputFile.is_open()) {  
        std::cerr << "Не вдалося відкрити вхідний файл." << std::endl;  
        return 1;  
    }
```

```
    // Зчитуємо x, y, v з файлу  
    inputFile >> x >> y >> z;  
    inputFile.close();
```

```
    // Виконуємо обчислення за допомогою функції  
    double result = s_calculation(x, y, z);  
    std::string binaryV = toBinary(z);
```

```
    std::ofstream    outputFile(outputFileName,    std::ios_base::app);    //
```

Відкриваємо файл для дозапису

```
    if (!outputFile.is_open()) {  
        std::cerr << "Не вдалося відкрити вихідний файл." << std::endl;
```

```
        return 1;
    }

    // Допишуємо результати у вихідний файл
    outputFile << "Результат s_calculation: " << result << std::endl;
    outputFile << "Число z у двійковому коді: " << binaryV << std::endl;

    outputFile.close();

    return 0;
}
```

Висновок

В цій лабораторній роботі я набув ґрунтовних вмінь і практичних навичок реалізації у Code::Blocks IDE мовою програмування C++ 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 символічної інформації.