

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.

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

ВИКОНАВ  
студент академічної групи КБ-24  
Жуковська Владислава

ПЕРЕВІРИВ  
ст. викладач кафедри кібербезпеки  
та програмного забезпечення  
Коваленко Анастасія Сергіївна

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

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.

<https://github.com/odorenskyi/>

## ВАРІАНТ 13

— ВХІДНИЙ ТЕКСТ - ВМІСТ ВХІДНОГО ТЕКСТОВОГО ФАЙЛУ —

Свята моя ненько, я вдячний Тобі,  
Що в хаті тепленько і сіль є, і хліб,  
За те, що ще можу свій хрест я нести,  
О Матінко Божа, спаси і прости.

— ЗАДАЧА 10.1 —

У вихідний текстовий файл записати:

- авторську інформацію: ім'я й прізвище розробника модуля, установа/організація, місто, країна, рік розробки;
- всі літери української абетки;
- повідомлення про наявність або відсутність помилок у словах вірша Генріха Акулова “Молитва” (наприклад, “типленько”, “ненька”) із вхідного файлу.

— ЗАДАЧА 10.2 —

У вихідний текстовий файл дописати:

- перелік Державних символів України згідно зі ст. 20 Конституції України;
- 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 з заголовкового файлу “ПрізвищеModule.h”:

- 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`.

## **Аналіз і постановка задач 10.1, 10.2 та 10.3**

### **ЗАДАЧА 10.1**

**У вхідний текстовий файл записати:**

- **авторську інформацію: ім'я й прізвище розробника модуля, установа/організація, місто, країна, рік розробки;**
- **всі літери української абетки;**
- **повідомлення про наявність або відсутність помилок у словах вірша Генріха Акулова "Молитва" (наприклад, "типленько", "ненька") із вхідного файлу.**

**Аналіз задачі 10.1:**

**Задача передбачає обробку текстового файлу, який містить вірш. Необхідно виконати наступні кроки:**

- 1. Зчитати вміст вхідного файлу.**
- 2. Визначити авторську інформацію. Ця інформація повинна бути надана розробником модуля і записана у вихідний файл у вказаному форматі.**
- 3. Згенерувати українську абетку. Потрібно створити рядок, що містить усі літери українського алфавіту.**
- 4. Перевірити наявність помилок у словах вірша. Для цього необхідно порівняти кожне слово з вхідного файлу з правильним написанням відповідних слів вірша "Молитва" Генріха Акулова.**
- 5. Сформувані повідомлення про наявність або відсутність помилок. Залежно від результатів перевірки, у вихідний файл потрібно записати відповідне повідомлення.**
- 6. Записати всю необхідну інформацію у вихідний текстовий файл.**

**Постановка задачі 10.1:**

**Розробити програму, яка:**

- 1. Приймає шлях до вхідного текстового файлу як вхідний параметр.**
- 2. Зчитує вміст вхідного файлу.**
- 3. Записує у вихідний текстовий файл (наприклад, з ім'ям output10\_1.txt):**
  - **Авторську інформацію у форматі: "Ім'я Прізвище, Установа/Організація, Місто, Країна, Рік".**

- Усі літери української абетки в одному рядку.
- Повідомлення про наявність або відсутність помилок у словах вірша "Молитва" порівняно з вмістом вхідного файлу. Наприклад: "У вірші знайдено помилки." або "Помилки у вірші не знайдено."

## **ЗАДАЧА 10.2**

**У вхідний текстовий файл дописати:**

- перелік Державних символів України згідно зі ст. 20 Конституції України;
- дату й час поточної інформації.

**Аналіз задачі 10.2:**

Задача передбачає дозапис інформації у вже існуючий вхідний текстовий файл. Необхідно виконати наступні кроки:

1. Отримати перелік Державних символів України. Згідно зі ст. 20 Конституції України, це Державний Прапор України, Державний Герб України та Державний Гімн України.
2. Отримати поточну дату та час. Необхідно використати системні функції для отримання актуальної дати та часу.
3. Дописати отриману інформацію у кінець вхідного текстового файлу.

**Постановка задачі 10.2:**

**Розробити програму, яка:**

1. Приймає шлях до вхідного текстового файлу як вхідний параметр.
2. Допише у кінець цього файлу:
  - Перелік Державних символів України: "Державний Прапор України, Державний Герб України, Державний Гімн України." (або кожен символ з нового рядка).
  - Поточну дату та час у форматі "ДД.ММ.РРРР ЧЧ:ММ:СС".

## **ЗАДАЧА 10.3**

Вхідні дані - числові значення  $x$ ,  $y$ ,  $z$  та натуральне число  $b$ . У вихідний текстовий файл дописати результати виконання функцій з заголовкового файлу "Prisvce.h":

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

### Аналіз задачі 10.3:

Задача передбачає використання зовнішнього заголовкового файлу та виконання певних операцій з числовими даними. Необхідно виконати наступні кроки:

1. Отримати вхідні числові дані:  $x$ ,  $y$ ,  $z$  (дійсні числа) та  $b$  (натуральне число).
2. Підключити заголовковий файл "Prisvce.h" з каталогу "Modules". Програма повинна мати доступ до цього файлу.
3. Викликати функцію `s_calculation` з заголовкового файлу, передавши їй аргументи  $x$ ,  $y$ ,  $z$ .
4. Отримати результат виконання функції `s_calculation`.
5. Перетворити натуральне число  $b$  у його двійкове представлення.
6. Дописати отримані результати у вихідний текстовий файл.

### Постановка задачі 10.3:

Розробити програму, яка:

1. Приймає чотири вхідні параметри: дійсні числа  $x$ ,  $y$ ,  $z$  та натуральне число  $b$ .
2. Підключає заголовковий файл `Modules.h`.
3. Викликає функцію `s_calculation(x, y, z)` з підключеного файлу та отримує результат.
4. Перетворює число  $b$  у його двійковий строковий еквівалент.
5. Записує у вихідний текстовий файл (наприклад, з ім'ям `output10_3.txt`):
  - Результат виконання функції `s_calculation` у форматі: "Результат `s_calculation(x, y, z)` = [значення]".
  - Двійкове представлення числа  $b$  у форматі: "Число  $b$  в двійковому коді: [двійкове значення]".

### Лістинг 10.1

```
void task10_1(const std::string& inputFilename, const std::string& outputFilename) {  
    // Internal helper functions  
    auto checkFileExistence = [](const std::string& filename) {  
        std::ifstream file(filename);  
        return file.good();  
    };  
};
```

```

auto readInputFile = [](const std::string& filename) {
    std::wifstream wif(filename);
    wif.imbue(std::locale(std::locale(), new std::codecvt_utf8<wchar_t>));

    std::wstring content;
    std::wstring line;

    while (std::getline(wif, line)) {
        content += line + L"\n";
    }

    return content;
};

auto writeToOutputFile = [](const std::string& filename, const std::wstring&
content) {
    std::wofstream wof(filename);
    wof.imbue(std::locale(std::locale(), new std::codecvt_utf8<wchar_t>));

    if (wof.is_open()) {
        wof << content;
        wof.close();
    }
};

// Check if input file exists before processing
if (!checkFileExistence(inputFilename)) {
    std::cout << "Помилка: Вхідний файл не знайдено!\n";
    return;
}

// Read input text
std::wstring inputText = readInputFile(inputFilename);

if (inputText.empty()) {
    std::cout << "Помилка: Вхідний файл порожній або не може бути прочитаний!\n";
    return;
}

std::wstring outputContent;

// Author information
outputContent += L"Авторська інформація:\n";
outputContent += L"Ім'я й прізвище розробника модуля: Жуковська Владислава\n";

```

```

outputContent += L"Установа/організація: ЦНТУ\n";
outputContent += L"Місто, країна: Кропивницький, Україна\n";
outputContent += L"Рік розробки: 2025\n\n";

// Ukrainian alphabet
outputContent += L"Всі літери української абетки:\n";
outputContent += L"а б в г г д е є ж з и і ї к л м н о п р с т у ф х ц ч ш щ ь
ю я\n\n";

// Check for errors in the poem
outputContent += L"Перевірка помилок у вірші:\n";

std::wstring poem = inputText;
std::vector<std::wstring> errors;
std::wstringstream wss(poem);
std::wstring word;

std::vector<std::wstring> correctWords = {
    L"Свята", L"моя", L"Ненько", L"я", L"вдячний", L"Тобі",
    L"Що", L"в", L"хати", L"тепленько", L"і", L"сіль", L"є", L"ї", L"хліб",
    L"За", L"те", L"що", L"ще", L"можу", L"свій", L"хрест", L"я", L"нести",
    L"О", L"Матінко", L"Божа", L"спаси", L"і", L"прости"
};

int wordIndex = 0;
while (wss >> word) {
    // Remove punctuation from the word
    for (int i = 0; i < word.length(); ++i) {
        if (std::ispunct(word[i], std::locale())) {
            word.erase(i--, 1);
        }
    }

    if (wordIndex < correctWords.size() && word != correctWords[wordIndex]) {
        errors.push_back(L"Помилка у слові: \"" + word + L"\", очікується \"" +
correctWords[wordIndex] + L"\"");
    }
    wordIndex++;
}

if (errors.empty()) {
    outputContent += L"Помилки у словах вірша не виявлено.\n\n";
} else {
    for (const auto& error : errors) {
        outputContent += error + L"\n";
    }
}

```

```

    }
    outputContent += L"\n";
}

// Write to output file
writeToOutputFile(outputFilename, outputContent);

    std::cout << "Завдання 10.1 виконано. Результати записані у файл " <<
outputFilename << "\n";
}

```

## Лістинг 10.2

```

void task10_2(const std::string& outputFilename) {
    // Internal helper functions
    auto appendToOutputFile = [](const std::string& filename, const std::wstring&
content) {
        std::wofstream wof(filename, std::ios::app);
        wof.imbue(std::locale(std::locale(), new std::codecvt_utf8<wchar_t>));

        if (wof.is_open()) {
            wof << content;
            wof.close();
        }
    };

    auto getCurrentDateTime = []() {
        time_t now = time(0);
        struct tm tstruct;
        char buf[80];

        localtime_s(&tstruct, &now);
        strftime(buf, sizeof(buf), "%d.%m.%Y %H:%M:%S", &tstruct);

        std::string str(buf);
        std::wstring wstr(str.begin(), str.end());
        return wstr;
    };

    std::wstring content;

    content += L"\n--- Державні символи України ---\n";
    content += L"Згідно зі статтею 20 Конституції України:\n";
    content += L"1. Державний Прапор України — стяг із двох рівновеликих
горизонтальних смуг синього і жовтого кольорів.\n";
}

```



```

        content += L"2. Великий Державний Герб України встановлюється з урахуванням
малого Державного Герба України та герба Війська Запорізького.\n";

        content += L"3. Головним елементом великого Державного Герба України є Знак
Княжої Держави Володимира Великого (малий Державний Герб України).\n";

        content += L"4. Державний Гімн України – національний гімн на музику М.
Вербицького зі словами, затвердженими законом.\n";

        content += L"5. Опис державних символів України та порядок їх використання
встановлюються законом.\n\n";

// Add timestamp
content += L"Дата і час дозапису інформації: " + getCurrentDateTime() + L"\n\n";

// Append to output file
appendToOutputFile(outputFilename, content);

        std::cout << "Завдання 10.2 виконано. Результати додані до файлу " <<
outputFilename << "\n";
}

```

### Лістинг 10.3

```

void task10_3(double x, double y, double z, unsigned int b) {
    std::ofstream outputFile("output.txt", std::ios::app);
    if (!outputFile.is_open()) {
        std::cout << "Помилка при відкритті вихідного файлу" << std::endl;
        return;
    }

    // Виклик функції s_calculation з аргументами x, y, z
    double result = s_calculation(x, y, z);
    outputFile << "Результат функції s_calculation(" << x << ", " << y << ", " << z
<< ") = " << result << std::endl;

    // Конвертація числа b у двійковий код
    outputFile << "Число " << b << " у двійковому коді: ";

    // Використовуємо бітові операції для виведення двійкового представлення
    if (b == 0) {
        outputFile << "0";
    } else {
        // Знаходимо кількість значущих бітів
        int bits = 0;
        unsigned int temp = b;
        while (temp > 0) {
            bits++;

```

```

        temp >>= 1;
    }

    // Виводимо біти від старшого до молодшого
    for (int i = bits - 1; i >= 0; i--) {
        outputFile << ((b >> i) & 1);
    }
}

outputFile << std::endl;
outputFile.close();
}

```

## Лістинг TestDriver

```

#include "ModulesZhukovsky.h"
#include <windows.h>
#include <iostream>
#include <limits>
#include <string>

// Функція для преобразования std::wstring в std::string
std::string toString(const std::wstring& wstr) {
    int size_needed = WideCharToMultiByte(CP_UTF8, 0, wstr.c_str(), -1, nullptr, 0,
    nullptr, nullptr);
    std::string str(size_needed, 0);
    WideCharToMultiByte(CP_UTF8, 0, wstr.c_str(), -1, &str[0], size_needed, nullptr,
    nullptr);
    return str;
}

int main() {
    // Установка кодировки UTF-8 для консоли
    SetConsoleOutputCP(CP_UTF8);
    SetConsoleCP(CP_UTF8);

    std::string inputFilename = "input.txt";
    std::string outputFilename = "output.txt";
    int choice;

    do {
        std::cout << "Оберіть завдання для виконання:\n"
            << "1 - Завдання 10.1\n"
            << "2 - Завдання 10.2\n"
            << "3 - Завдання 10.3\n"
            << "0 - Вихід\n"
            << "Ваш вибір: ";
        while (!(std::cin >> choice)) {
            std::cin.clear();
            std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
            std::cout << "Помилка! Введіть коректне значення: ";
        }
    }
}

```

```

switch (choice) {
    case 1:
        std::cout << "Виконується завдання 10.1...\n";
        task10_1(inputFilename, outputFilename);
        std::cout << "Завдання 10.1 завершено.\n";
        break;

    case 2:
        std::cout << "Виконується завдання 10.2...\n";
        task10_2(outputFilename);
        std::cout << "Завдання 10.2 завершено.\n";
        break;

    case 3: {
        std::cout << "Виконується завдання 10.3...\n";
        double x, y, z;
        int b;

        std::cout << "Завдання 10.3 - введіть дані для обробки:\n";

        std::cout << "Введіть значення x: ";
        while (!(std::cin >> x)) {
            std::cin.clear();
            std::cin.ignore(std::numeric_limits<std::streamsize>::max(),
'\n');

            std::cout << "Помилка! Введіть коректне значення для x: ";
        }

        std::cout << "Введіть значення y: ";
        while (!(std::cin >> y)) {
            std::cin.clear();
            std::cin.ignore(std::numeric_limits<std::streamsize>::max(),
'\n');

            std::cout << "Помилка! Введіть коректне значення для y: ";
        }

        std::cout << "Введіть значення z: ";
        while (!(std::cin >> z)) {
            std::cin.clear();
            std::cin.ignore(std::numeric_limits<std::streamsize>::max(),
'\n');

            std::cout << "Помилка! Введіть коректне значення для z: ";
        }

        std::cout << "Введіть додатне ціле число b: ";
        while (!(std::cin >> b) || b <= 0) {
            std::cin.clear();
            std::cin.ignore(std::numeric_limits<std::streamsize>::max(),
'\n');

            std::cout << "Помилка! Введіть додатне ціле число для b: ";
        }

        try {
            task10_3(x, y, z, b);

```

```

        } catch (const std::exception& e) {
            std::cerr << "Помилка під час виконання завдання 10.3: " <<
e.what() << "\n";
        }
        std::cout << "Завдання 10.3 завершено.\n";
        break;
    }

    case 0:
        std::cout << "Вихід з програми.\n";
        break;

    default:
        std::cout << "Невірний вибір. Будь ласка, спробуйте ще раз.\n";
    }
} while (choice != 0);

std::cout << "Ви успішно завершили виконання програми.\n";

return 0;
}

```

## Лістинг ModulesZhukovsky.h

```

#ifndef MODULESZHUKOVSKY_H
#define MODULESZHUKOVSKY_H

double s_calculation(double x, double y, double z);
void developer_info();

//lab 9
#include <string>
char calculateRating(int test1, int test2, int exam);
std::string convertShoeSize(double size_cm);
int binaryBitCount(unsigned int N);

//lab 10
void task10_1(const std::string& inputFilename, const std::string& outputFilename);
void task10_2(const std::string& outputFilename);
void task10_3(double x, double y, double z, unsigned int b);

#endif // MODULESZHUKOVSKY_H

```

## Висновок до лабораторної роботи.

Виконання цієї лабораторної роботи дозволило глибше зануритися в процес розробки програм, що працюють з текстовими файлами мовою C++. Основна мета полягала у створенні та доповненні вхідного текстового файлу відповідно до заданих умов, що включають як текстову обробку, так і числові обчислення.

Протягом роботи було послідовно реалізовано всі етапи: створення файлу, внесення інформації про розробника, перевірка тексту на наявність помилок, дописування української абетки, державної символіки, поточної дати та часу. Це дозволило закріпити навички роботи з файлами у різних режимах – читання, запис, дописування – а також сформувати вміння ефективно обробляти символічну та рядкову інформацію.

Особливий інтерес становила робота з аналізом віршованого тексту. Потрібно було не лише розпізнати слова, а й виявити можливі орфографічні помилки. Це завдання поєднує в собі навички з обробки рядків, логічного аналізу та елементарних алгоритмів перевірки тексту. Цікавою особливістю було те, що об'єктом перевірки став текст українською мовою, що вимагає врахування національних мовних особливостей, включаючи специфіку алфавіту, літер з діакритичними знаками та правила правопису. Такий досвід є надзвичайно корисним з огляду на зростаючу роль обробки природної мови у сучасному програмуванні.

Додавання інформації про державні символи України згідно зі статтею двадцятою Конституції України мало не лише практичне, а й виховне значення, формуючи вміння звертатися до офіційних документів і працювати з нормативними текстами. Також важливим було доповнення файлу поточними датою і часом, що вимагало залучення стандартних бібліотек мови C++ для взаємодії з системним часом. Це є базовим елементом для формування логів, маркування змін і реалізації функціоналу, пов'язаного з відстеженням активностей у більш складних програмних продуктах.

На заключному етапі роботи було реалізовано математичні обчислення з використанням заголовкового файлу з іменем користувача. У рамках цього етапу були викликані функції з аргументами, проведені обчислення та збережені результати у текстовому файлі. Додатково значення натурального числа було переведено в двійкову систему числення, що дозволило закріпити навички роботи з системами числення та перетворення чисел у нестандартні формати для збереження або подальшої обробки.

Загалом дана лабораторна робота охопила кілька важливих тем, включаючи роботу з файлами, обробку символічної інформації, аналіз текстів, виконання арифметичних дій і представлення чисел у двійковому вигляді. Це сприяло розвитку аналітичного мислення, алгоритмічної грамотності, навичок структурованого підходу до

розв'язання прикладних завдань. Робота має практичне значення, оскільки подібні завдання зустрічаються в реальних умовах розробки програмного забезпечення – при створенні логів, аналізі даних, генерації звітів, роботі з базами даних, а також в системах, що працюють з текстами природною мовою. Таким чином, виконання цього завдання сприяло формуванню базових, але надзвичайно важливих навичок програмування, які є основою професійного зростання фахівця з інформаційних технологій.