

Міністерство освіти і науки України  
Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет

ЗВІТ  
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ №9  
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  
Олефіров Г.Є.  
<https://github.com/GlibOlefirov>

ПЕРЕВІРИВ  
ст. викладач кафедри кібербезпеки  
та програмного забезпечення  
Коваленко А.С.

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

**Мета:** 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 символної інформації.

### **ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ**

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

### **Варіант № 18**

#### **Аналіз і постановка задачі 10.1**

У задачі 10.1 потрібно зчитати слово з вхідного текстового файлу та визначити кількість приголосних літер у ньому. У створений вихідний файл слід записати авторську інформацію (ПІБ, організація, місто, країна, рік), результати аналізу слова, а також віршоване повідомлення. Необхідно перевірити, чи входить зчитане слово до тексту вірша, і вивести відповідне повідомлення. Програма має забезпечити правильне зчитування, обробку тексту українською мовою та коректне форматування вихідного файлу.

### **Аналіз і постановка задачі 10.2**

У задачі 10.2 необхідно проаналізувати вхідний текстовий файл. Потрібно визначити кількість символів, що містяться у цьому файлі, отримати поточну дату й час, а потім дописати у вхідний файл інформацію про кількість символів й поточну дату. Таким чином, дані у файлі доповнюються новим змістом, а не замінюються. Уся інформація має бути записана українською мовою.

### **Аналіз і постановка задачі 10.3**

У задачі 10.3 потрібно опрацювати чотири натуральні числа:  $x$ ,  $y$ ,  $z$  та  $b$ . Для чисел  $x$ ,  $y$ ,  $z$  необхідно викликати функцію `s_calculation(x, y, z)`, яка реалізована у заголовковому файлі `ModulesOlefirov.h`. Результат обчислення слід дописати у вихідний текстовий файл. Крім того, число  $b$  потрібно перевести у двійкову форму та також записати у цей файл. Уся інформація має бути записана у зрозумілому вигляді та українською мовою.

### **Результат тестування**

=== Тестування завдань 10.1 10.2 10.3 ===

Тест 10.1: аналіз слів

Тест 10.1 #1 пройдено

Тест 10.1 #2 пройдено

Тест 10.1 #3 пройдено

Тест 10.2: допис інформації у файл

Тест 10.2 #1 пройдено

Тест 10.2 #2 пройдено

Тест 10.2 #3 пройдено

Тест 10.3: обчислення S і двійкове представлення

Тест 10.3 #1 пройдено

Тест 10.3 #2 пройдено

Тест 10.3 #3 пройдено

Усі тести пройдені успішно!

## Додаток

### Лістинг TestDriver

```
#include <iostream>

#include "Modules01efirov.h"

#include <fstream>
#include <string>
#include <cassert>
#include <locale.h>

using namespace std;

// Функція для перевірки чи файл містить потрібний текст
bool fileContains(const string& filename, const string& text) {
    ifstream file(filename);
    string line;
    while (getline(file, line)) {
        if (line.find(text) != string::npos) return true;
    }
    return false;
}
```

```
int main() {
    setlocale(LC_ALL, "ukr");
    cout << "=== Тестування завдань 10.1 10.2 10.3 ===\n";
    // ----- Тести для задачі 10.1 -----
    cout << "\nТест 10.1: аналіз слів\n";
    string words[] = {"щасливий", "любов", "вiра"};
    for (int i = 0; i < 3; ++i) {
        string input = "input_task_10.1_" + to_string(i+1) + ".txt";
        string output = "output_task_10.1_" + to_string(i+1) + ".txt";
        ofstream f(input);
        f << words[i];
        f.close();
        processText(input, output);
        assert(fileContains(output, "Кількість приголосних"));
        assert(fileContains(output, "Слово"));
        cout << "Тест 10.1 #" << i+1 << " пройдено \n";
    }
    // ----- Тести для задачі 10.2 -----
    cout << "\nТест 10.2: допис інформації у файл\n";
    string initialTexts[] = {"", "Тест!", "Не придумав тексту для перевірки"};
    for (int i = 0; i < 3; ++i) {
        string filename = "output_task_10.2_" + to_string(i+1) + ".txt";
        ofstream f(filename);
        f << initialTexts[i];
        f.close();
        appendInfoToFile(filename);
        assert(fileContains(filename, "Кількість символів"));
        assert(fileContains(filename, "Дата дозапису"));
        cout << "Тест 10.2 #" << i+1 << " пройдено \n";
    }
}
```

```

}

// ----- Тести для задачі 10.3 -----

cout << "\nТест 10.3: обчислення S і двійкове представлення\n";

struct TestData {
double x, y, z;
int b;
} testData[] = {
{1, 2, 3, 5}, // звичайний випадок
{0, 1, 2, 15}, // x=0
{1, 2, -1, 7} // z<0 (некоректне z)
};

for (int i = 0; i < 3; ++i) {
string filename = "output_task_10.3_" + to_string(i+1) + ".txt";
processTask10_3(testData[i].x, testData[i].y, testData[i].z, testData[i].b,
filename);
assert(fileContains(filename, "s_calculation"));
assert(fileContains(filename, "Число b у двійковій системі"));
cout << "Тест 10.3 #" << i+1 << " пройдено \n";
}

cout << "\nУсі тести пройдені успішно! \n";
return 0;
}

```

### Лістинг ModulesOlefirov

```

#include <iostream>
#include <math.h>
#include <fstream>
#include <string>
#include <cctype>

```

```

#include <ctime>

#include <sstream>

#include <bitset>

using namespace std;

// Функція для обчислення S
double calculate_S(double x, double y, double z) {
    // Перевірка на некоректне z
    if (z < 0) return -1;
    // Обчислення знаменника
    double denominator = y + 4 * (pow(x, 3) + cos(z));
    // Перевірка на ділення на 0 або від'ємне значення в корені
    if (denominator <= 0) return -1;
    // Обчислення чисельника
    double numerator = 2 * M_PI * sqrt(0.5 * z);
    // Рахує значення виразу під коренем
    double root_part = sqrt(numerator / denominator);
    // Повертає результат виразу
    return pow(z + y + z, 2) - root_part;
}

void calculateGasPayment() {
    double volume = 0.0;
    cout << "Введіть об'єм газу: ";
    cin >> volume ;
    double cost = 0.0;
    if (volume <= 208) {
        cost = volume * 1.299;
    }
    else if (volume <= 500) {
        cost = 208 * 1.299 + (volume - 208) * 1.788;
    }
}

```

```
}  
else {  
cost = 208 * 1.299 + (500 - 208) * 1.788 + (volume - 500) * 3.645;  
}  
cout << "Ціна за використання газу: " << cost << endl << endl ;  
}  
void findShoeSize(){  
double size_cm=0;  
cout << "Введіть розмір у сантиметрах: ";  
cin >> size_cm;  
int size_mm = size_cm * 10;  
switch(size_mm){  
case 205:  
cout << "Україна: -, Великобританія: 1\n";  
break;  
case 210:  
cout << "Україна: -, Великобританія: 1.5\n";  
break;  
case 215:  
cout << "Україна: -, Великобританія: 2\n";  
break;  
case 220:  
cout << "Україна: -, Великобританія: 2.5\n";  
break;  
case 225:  
cout << "Україна: -, Великобританія: 3\n";  
break;  
case 230:  
cout << "Україна: 35, Великобританія: 4\n";
```



```
break;

case 235:
cout << "Україна: 36, Великобританія: 4.5\n";
break;

case 240:
cout << "Україна: 36.5, Великобританія: 5\n";
break;

case 245:
cout << "Україна: 37, Великобританія: 5.5\n";
break;

case 250:
cout << "Україна: 38, Великобританія: 6\n";
break;

case 255:
cout << "Україна: 38/39, Великобританія: 6.5\n";
break;

case 260:
cout << "Україна: 40.5, Великобританія: 7.5\n";
break;

case 265:
cout << "Україна: 41, Великобританія: 8\n";
break;

case 270:
cout << "Україна: 41.5, Великобританія: 8.5\n";
break;

case 275:
cout << "Україна: 42, Великобританія: 9\n";
break;

case 280:
```

```
cout << "Україна: 42/43, Великобританія: 9.5\n";  
break;  
case 285:  
cout << "Україна: 43, Великобританія: 10\n";  
break;  
case 290:  
cout << "Україна: 44, Великобританія: 11\n";  
break;  
case 295:  
cout << "Україна: 45, Великобританія: 11.5\n";  
break;  
case 300:  
cout << "Україна: 46, Великобританія: 12\n";  
break;  
case 305:  
cout << "Україна: 47, Великобританія: 12.5\n";  
break;  
case 310:  
cout << "Україна: 47.5, Великобританія: 13\n";  
break;  
case 315:  
cout << "Україна: 48, Великобританія: 14\n";  
break;  
case 320:  
cout << "Україна: 48.5, Великобританія: 14.5\n";  
break;  
default:  
cout << "Такого розміру не існує\n";  
}
```

```

}

void processNumber() {
    unsigned int N;

    cout << endl << "Введіть натуральне число N (0..21359010): ";

    cin >> N;

    int count0 = 0, count1 = 0;

    unsigned int temp = N;

    // Підрахунок кількості нулів та одиниць у двійковому представленні
    while (temp > 0) {
        (temp % 2 == 0) ? count0++ : count1++;
        temp /= 2;
    }

    // Отримання 11-го біта справа
    bool bit11 = (N >> 11) & 1;

    // Вивід результату
    cout << "Результат: " << (bit11 ? count0 : count1) << endl;
}

// Перевірка на приголосні
bool isConsonant(char ch) {
    ch = tolower(ch);

    string cons = "бвггджзйклмнпрстфхцчщ";

    for (char c : cons) {
        if (ch == c) return true;
    }

    return false;
}

// Завдання 10.1
void processText(const string& inputFile, const string& outputFile) {
    ifstream in(inputFile);

```

```

ofstream out(outputFile);

if (!in.is_open() || !out.is_open()) {
    cout << "Помилка відкриття файлів!" << endl;
    return;
}

string word;

in >> word;

in.close();

int count = 0;

for (char ch : word) {
    if (isConsonant(ch)) count++;
}

string poem = "Про себе не кажи недобрих слів, "
"Бо має сказане таємну силу. "
"Кажі: «Я сильний, впевнений, щасливий!» "
"I буде саме так, як ти хотів!";

bool found = (poem.find(word) != string::npos);

out << "Автор: Олефіров Гліб, ЦНТУ, Кропивницький, Україна, 2025\n";
out << "Слово: " << word << "\n";
out << "Кількість приголосних: " << count << "\n";
out << "Слово " << (found ? "є" : "немає") << " у вірші Іващенко.\n";
out.close();
}

// Завдання 10.2

void appendInfoToFile(const string& filename) {
    ifstream inFile(filename);

    int charCount = 0;

    char ch;

    while (inFile.get(ch)) charCount++;

```

```

inFile.close();

time_t now = time(0);

tm* localTime = localtime(&now);

stringstream dateStream;

dateStream << "Дата дозапису: "

<< (localTime->tm_mday < 10 ? "0" : "") << localTime->tm_mday << "."

<< (localTime->tm_mon + 1 < 10 ? "0" : "") << (localTime->tm_mon + 1) << "."

<< (1900 + localTime->tm_year) << " "

<< (localTime->tm_hour < 10 ? "0" : "") << localTime->tm_hour << ":"

<< (localTime->tm_min < 10 ? "0" : "") << localTime->tm_min << ":"

<< (localTime->tm_sec < 10 ? "0" : "") << localTime->tm_sec;

ofstream outFile(filename, ios::app);

outFile << "\nКількість символів: " << charCount << endl;

outFile << dateStream.str() << endl;

outFile.close();

}

void processTask10_3(double x, double y, double z, int b, const string&
outputFilename) {

double result = calculate_S(x, y, z);

bitset<8> binaryB(b);

ofstream outFile(outputFilename, ios::app);

outFile << "\n=== Результат задачі 10.3 ===\n";

outFile << "s_calculation(x, y, z) = " << result << endl;

outFile << "Число b у двійковій системі: " << binaryB << endl;

outFile.close();

}

```

## Висновок

1. Засвоєно синтаксис оголошення одномірних і двовимірних масивів у C++, що дозволяє ефективно працювати з наборами даних.

2. Вивчено принципи індексування масивів, що є критично важливими для доступу до елементів.
3. Ознайомлено з функціями `<cstring>` для обробки рядків, які спрощують роботу з символьними масивами.
4. Зрозуміло, як працює функція `strcpy` для копіювання рядків, що дозволяє уникати помилок при передачі текстових значень.
5. Навченося використовувати `strlen` для визначення довжини рядка, що є корисним при перевірках і циклічній обробці.
6. Функція `strcmp` виявилась зручною для порівняння текстових значень у логічних умовах.
7. Навички роботи з `strcat` дали змогу ефективно конкатенувати (з'єднувати) рядки.
8. Опановано використання структур `struct` як засобу організації даних різних типів.
9. Вивчено різницю між масивами й структурами: масиви — однорідні, структури — гетерогенні.
10. Опановано синтаксис вкладених структур, що дозволяє будувати складніші моделі даних.
11. Розглянуто оголошення змінної типу структура через заголовковий файл, що сприяє модульності коду.
12. Вивчено множинний тип змінних і операції над множинами, як-от перевірка наявності, додавання й вилучення.
13. Опановано явне та неявне перетворення типів у C++, що є важливим для запобігання помилкам типів.
14. Навченося створювати об'єкти `ofstream` для запису у файл, що дозволяє зберігати результати виконання програм.
15. Вивчено режими відкриття потоків: `in`, `out`, `app`, `trunc`, `ate`, що допомагає контролювати обробку файлів.
16. Зрозуміло призначення функцій `open()`, `close()`, `eof()` у роботі з потоками введення/виведення.
17. Опановано створення бібліотеки `libModulesПрізвище.a`, що сприяє повторному використанню коду.

18. Навченося реалізовувати модулі у вигляді функцій із заданим інтерфейсом.
19. Отримано навички інтеграції модулів у статичну бібліотеку та підключення до іншого проєкту.
20. Зрозуміло важливість тест-драйвера для перевірки коректності програмних рішень.
21. Вивчено структуру тест-кейсу з полями: вхідні дані, дії, очікуваний результат.
22. Навченося створювати автоматизоване unit-тестування для оцінки правильності виконання функцій.
23. Розроблено кілька тест-сценаріїв (тест-сютів) для задач 10.1–10.3.
24. Реалізовано логіку логування результатів тестування у файл, що дозволяє зберігати історію перевірок.
25. Отримано досвід побудови архітектури програмного модуля.
26. Розроблено власні проєктні рішення для кожної задачі.
27. Проаналізовано вхідні текстові файли як джерело даних для модулів.
28. Побудовано детальне проєктування функціоналу кожної задачі.
29. Опановано створення нових проєктів у Code::Blocks та налаштування опцій компілятора.
30. Навченося використовувати заголовковий файл для підключення функцій із бібліотеки.
31. Виконано компіляцію та налагодження бібліотеки та драйвера.
32. Забезпечено передачу імен файлів як аргументів у функції.
33. Реалізовано зчитування та порівняння вмісту вихідного файлу з очікуваним результатом.
34. Забезпечено механізм виводу статусу тест-кейса: passed / failed.
35. Вдосконалено навички роботи зі змінними різних типів.
36. Усвідомлено важливість модульного тестування для покращення якості ПЗ.
37. Підвищено рівень володіння синтаксисом мови C++.
38. Вивчено тип union як спосіб економії пам'яті.
39. Реалізовано приклади видалення підрядків із файлів.

40. Проаналізовано спосіб пошуку натурального числа у текстовому файлі.
41. Здійснено використання циклів та умов у функціях обробки файлів.
42. Вивчено, як визначити обсяг текстового файлу, що містить значення  $\pi$  з 106 цифрами.
43. Використано `sizeof` для визначення розміру типу/виразу.
44. Пояснено відмінності між текстовими та бінарними потоками.
45. Розроблено README.md і структуру каталогів згідно з методичкою.
46. Завантажено виконану роботу до GitHub, що сприяє навичкам роботи з системами контролю версій.
47. Всі проекти задокументовано відповідно до стандарту ДСТУ 3008:2015.
48. У звіті чітко вказано цілі, хід виконання, проблеми й шляхи їх вирішення.
49. Досягнуто мети лабораторної роботи: сформовано вміння розробляти модулі, тестувати та працювати з файлами.
50. Отримано практичні навички командної та модульної розробки, що є важливими в сучасному програмуванні.