

Міністерство освіти і науки України  
Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення  
Дисципліна: БМТП

**Лабораторна робота №10**  
**РЕАЛІЗАЦІЯ СТАТИЧНИХ БІБЛІОТЕК МОДУЛІВ ЛІНІЙНИХ**  
**ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ**

Виконав: студент групи КН-22  
Фідря М.О.  
Перевірів: викладач  
Собінов Олександр Георгійович

Кропивницький 2023

**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:**

1. Реалізувати програмні модулі розв'язування задач 10.1–10.3 як складові статичної бібліотеки libModulesПрізвище.a
2. Реалізувати тестовий драйвер автоматизованої перевірки програмних модулів розв'язування задач 10.1–10.3.

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

ВАРІАНТ 1

— 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 символів у вхідному файлі;
- повідомлення, чи є у вхідному файлі слова “програма”, “модуль”, “студент”, “програміст” (в програмі слід реалізувати розрізнення слів “модуль” та “модульчик”, “студент” та “студентка”, “модуль” та “модульне” тощо).

— ЗАДАЧА 10.2 —

У вхідний текстовий файл дописати:

- кількість цифр 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 ModulesПрізвище.h `s_calculation` з аргументами  $x$ ,  $y$ ,  $z$ ;
- число  $b$  у двійковому коді.

**Аналіз і постановка задач 10.1-3:**

Після того як користувач вводить довільний текст у файлі "input.txt" програма підраховує кількість цифр у ньому, дату та час запису і дозаписує цю інформацію у файл "input.txt". Якщо користувач ще раз записує щось у

файл "input.txt" то програма перезаписує кількість цифр у ньому, дату та час дозапису.

Користувач вводить довільні значення у консолі  $x$ ,  $y$ ,  $z$  та натуральне число  $b$ , програма підраховує формулу  $x + y - z$  за допомогою функції `s_calculation` та виводить число  $b$  у двійковій системі,

Результати функції `s_calculation` та число  $b$  записуються у файл "output.txt". У файл "output.txt" також записується авторська інформація: ім'я й прізвище розробника, організація, місто, країна, рік розробки, кількість символів у файлі "input.txt" та повідомлення, чи є у файлі "input.txt" слова "program" "student" "module" "programer".

Створивши потрібні файли `main.cpp`, `library.cpp`, `library.h` та `input.txt` почав виконувати завдання.

### Алгоритм:

Початок

1. Зчитати вміст файлу "input.txt" в змінну `inputText`.
2. Закрити файл "input.txt".
3. Порахувати кількість цифр у змінній `inputText` та зберегти результат у змінну `digitCount`.
4. Отримати поточну дату та час і зберегти у змінну `dateTime`.
5. Створити рядок `fileInfo` зі значеннями "Digit count: ", `digitCount`, "\nDate and time of update: ", `dateTime`.
6. Перевірити, чи містить змінна `inputText` слова "program", "student", "module", "programer".
  - 6.1. Якщо містить, додати в рядок `fileInfo` повідомлення "Contains word: " та слово.
7. Відкрити файл "input.txt" для допису.
8. Дописати рядок `fileInfo` в кінець файлу "input.txt".
9. Запитати користувача про значення  $x$ ,  $y$ ,  $z$  та зберегти їх у відповідні змінні.
10. Обчислити результат функції `s_calculation` з використанням змінних  $x$ ,  $y$ ,  $z$  та зберегти у змінну `result`.
11. Запитати користувача про натуральне число  $b$  та зберегти його у змінну `b`.
12. Відкрити файл "output.txt" для запису.

13. Записати авторську інформацію, результат функції `s_calculation` та число `b` у двійковій системі у файл "output.txt".

14. Закрити файл "output.txt".

Кінець

*Лістинг library.cpp:*

```
#include "library.h"
#include <fstream>
#include <sstream>
#include <ctime>
#include <iostream>
#include <bitset>

int countDigits(const std::string& text) {
    int digitCount = 0;
    for (char c : text) {
        if (isdigit(c)) {
            digitCount++;
        }
    }
    return digitCount;
}

std::string getCurrentDateTime() {
    std::time_t now = std::time(nullptr);
    char buffer[20];
    struct tm timeinfo;
    localtime_s(&timeinfo, &now);
    std::strftime(buffer, sizeof(buffer),
"%Y-%m-%d %H:%M:%S", &timeinfo);
    return buffer;
}

void appendToFile(const std::string& fileName,
const std::string& content) {
    std::ofstream file(fileName, std::ios::app);
    if (file.is_open()) {
```

```

        file << content;
        file.close();
    }
    else {
        std::cout << "Unable to open file: " <<
fileName << std::endl;
    }
}

void processInputFile(const std::string&
inputFileName, const std::string& outputFileName) {
    std::ifstream inputFile(inputFileName);
    if (!inputFile.is_open()) {
        std::cout << "Unable to open input file: "
<< inputFileName << std::endl;
        return;
    }

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

    int digitCount = countDigits(inputText);
    std::string dateTime = getCurrentDateTime();
    std::string fileInfo = "Digit count: " +
std::to_string(digitCount) + "\n";
    fileInfo += "Date and time of update: " +
dateTime + "\n";

    // Check if the input text contains specific
words
    std::string words[] = { "program", "student",
"module", "programer" };
    for (const std::string& word : words) {
        if (inputText.find(word) !=
std::string::npos) {

```

```

        fileInfo += "Contains word: " + word +
"\n";
    }
}

appendToFile(inputFileName, fileInfo);

int x, y, z;
std::cout << "Enter values for x, y, and z: ";
std::cin >> x >> y >> z;

int result = s_calculation(x, y, z);

int b;
std::cout << "Enter a natural number b: ";
std::cin >> b;

std::ofstream outputFile(outputFileName);
if (outputFile.is_open()) {
    outputFile << "Author: Fidria Maksim" <<
std::endl;
    outputFile << "Organization: CNTU" <<
std::endl;
    outputFile << "City: Kropyvnytski" <<
std::endl;
    outputFile << "Country: Ukraine" <<
std::endl;
    outputFile << "Year of development: 2023"
<< std::endl;
    outputFile << "Input file character count:
" << inputText.size() << std::endl;
    outputFile << "Result of s_calculation: "
<< result << std::endl;
    outputFile << "Number b in binary: " <<
std::bitset<32>(b).to_string() << std::endl;
    outputFile.close();
}
else {

```

```

        std::cout << "Unable to open output file:
" << outputFileFileName << std::endl;
    }
}

int s_calculation(int x, int y, int z) {
    int result = x + y - z;
    return result;
}

```

*Лістинг library.h:*

```

#ifndef LIBRARY_H
#define LIBRARY_H

#include <string>

int countDigits(const std::string& text);
std::string getCurrentDateTime();
void appendToFile(const std::string& fileName,
const std::string& content);
void processInputFile(const std::string&
inputFileName, const std::string& outputFileFileName);

int s_calculation(int x, int y, int z);

#endif // LIBRARY_H

```

*Лістинг main.cpp:*

```

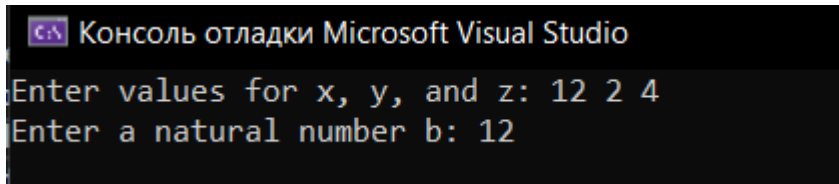
#include "library.h"

int main() {
    processInputFile("input.txt", "output.txt");
    return 0;
}

```

*Результат виконання програми:*

Додавши довільний текст <This is a text file for testing the operation of the program and the module. It contains several sentences that mention students and programmers. Some lines contain the words program module student and programmer . Students study various programming modules. Programmers develop programs for various tasks. In this text there is the word "programmer", but there is no word "program". Let this text be the input for verification.> до файлу "input.txt" запусив програму.



```
Консоль отладки Microsoft Visual Studio
Enter values for x, y, and z: 12 2 4
Enter a natural number b: 12
```

Ввів довільні числа та закінчив програму.

*Файл input.txt:*

This is a text file for testing the operation of the program and the module.

It contains several sentences that mention students and programmers.

Some lines contain the words program module student and programmer .

Students study various programming modules.

Programmers develop programs for various tasks.

In this text there is the word "programmer", but there is no word "program".

Let this text be the input for verification.

Digit count: 0

Date and time of update: 2023-06-13 13:14:58

Contains word: program

Contains word: student

Contains word: module

*Створився файл output.txt:*

Author: Fidria Maksim

Organization: CNTU

City: Kropyvnytski







кількості цифр у тексті, отримання поточної дати та часу, обчислення формули `s_calculation` та виведення числа `b` у двійковій системі.

У результаті цих завдань я практично використав різноманітні концепції та функції C++, що дозволило мені отримати більш глибоке розуміння мови та навичок програмування.

## ВІДПОВІДЬ НА КОНТРОЛЬНІ ЗАПИТАННЯ

1. Яке призначення та синтаксис запису блоку-контроля `try - throw - catch` у мові програмування C++?

Блок-контроль `try - throw - catch` у мові програмування C++ використовується для обробки винятків (exceptions). Він дозволяє забезпечити структуровану обробку помилок або непередбачуваних ситуацій, що виникають під час виконання програми.

```
try {  
    // Код, який потенційно може викликати виняток  
} catch (Тип_винятку_1) {  
    // Обробка винятку типу Тип_винятку_1  
} catch (Тип_винятку_2) {  
    // Обробка винятку типу Тип_винятку_2  
} catch (...) {  
    // Обробка будь-якого іншого винятку  
}
```

2. Наведіть приклад опису й використання міжмодульної змінної.

Міжмодульна змінна - це змінна, яка оголошується у одному модулі (файлі) програми і використовується в іншому модулі. Вона дозволяє обмінювати даними між різними модулями програми.

Ось приклад опису та використання міжмодульної змінної:

```
// module1.cpp  
// Оголошення міжмодульної змінної  
extern int globalVariable;  
  
void doSomething() {  
    // Використання міжмодульної змінної  
    globalVariable = 42;  
}
```

```

// module2.cpp
// Оголошення міжмодульної змінної
extern int globalVariable;

void printValue() {
    // Використання міжмодульної змінної
    cout << "The value of globalVariable is: " <<
globalVariable << endl;
}

// main.cpp
#include <iostream>
using namespace std;

// Оголошення міжмодульної змінної
int globalVariable;

int main() {
    // Використання міжмодульної змінної
    globalVariable = 10;

    // Виклик функцій з модулів
    doSomething();
    printValue();

    return 0;
}

```

3. Яку область видимості матимуть об'єкти (змінні, типи, константи тощо), описані в тілі функції main C++?

Об'єкти (змінні, типи, константи тощо), описані в тілі функції main у мові програмування C++, матимуть локальну область видимості. Це означає, що ці об'єкти будуть доступні лише всередині функції main і не будуть видимі в інших функціях або модулях програми. Локальні об'єкти, оголошені в тілі функції main, будуть існувати протягом виконання функції main. Вони створюються при вході до функції та знищуються при виході з функції. Таким чином, їх час життя обмежений виконанням функції main.

Область видимості локальних об'єктів включає блок коду, в якому вони оголошені. Вони будуть видимі лише всередині цього блоку та будуть недоступні поза ним.

#### 4. Здійснить порівняльний аналіз змінної типу `enum` та масиву.

Змінна типу `enum` і масив - це дві різні концепції у мові програмування C++. Розглянемо їх порівняльний аналіз:

Семантика:

- `enum`: Тип `enum` дозволяє створювати набір іменованих констант, які можна використовувати для представлення стану або категорії. Кожен елемент `enum` має унікальне ім'я та асоційоване з ним цілочисельне значення.
- Масив: Масив є структурою даних, що містить фіксовану кількість елементів одного типу, які розміщуються в послідовному порядку в пам'яті.

Декларація та використання:

- `enum`: `enum` оголошується за допомогою ключового слова `enum` і може містити список іменованих констант. Значення кожної константи встановлюється за замовчуванням, починаючи з 0 і збільшуючись на 1 для кожної наступної константи. Змінна типу `enum` може містити одне зі значень, які оголошені у типі `enum`.
- Масив: Масив оголошується за допомогою вказівника на його перший елемент та квадратних дужок, що вказують на його розмір. Кожен елемент масиву може містити значення будь-якого типу, включаючи примітивні типи даних, вказівники, структури, класи тощо. Елементи масиву можуть бути доступні по їхніх індексах.

Розмір:

- `enum`: Розмір `enum` визначається кількістю елементів, оголошених у типі `enum`. Кожен елемент займає пам'ять, необхідну для збереження цілочисельного значення.

- Массив: Розмір масиву визначається кількістю його елементів та розміром кожного елементу. Пам'ять, виділена для масиву, залежить від його розміру та типу елементів.

Доступ до елементів:

- enum: Елементи enum можна використовувати за допомогою їхніх імен. Наприклад, `enum Color { RED, GREEN, BLUE }; Color myColor = RED;`. Значення елементів можна порівнювати між собою, але не можна звертатися до них за допомогою індексів.
- Массив: Доступ до елементів масиву здійснюється за допомогою індексів. Наприклад, `int myArray[3] = {10, 20, 30}; int value = myArray[1];`. Елементи масиву можна порівнювати між собою та використовувати за допомогою індексів.

Отже, enum та масив мають відмінну семантику, декларацію та використання, а також різні способи доступу до їхніх елементів. Вибір між ними залежить від конкретного використання та потреб програми.