

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 РОЗГАЛУЖЕНИХ
ТА ІТЕРАЦІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

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

ВИКОНАВ
студент академічної групи КН-24
Булюкін В. Ю.
<https://github.com/kuroshi-dev>

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

17 Варіант

Тема: 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 10.1

У вихідний текстовий файл записати:

1. Авторську інформацію:
 - ім'я та прізвище розробника модуля;
 - установа/організація;
 - місто, країна;
 - рік розробки;
2. Кількість голосних літер у слові з вхідного файла;
3. Повідомлення, чи є слово із вхідного файлу в наступній краплиці Віталія Іващенко:

До щастя не пускає лінощів орава.
 У чім воно ніхто не знає до пуття.
 Навчитись радісно робити кожну справу
 Найперше правило *щасливого життя*.

Завдання 10.2

У вхідний текстовий файл дописати:

1. кількість приголосних літер у ньому, дату й час до запису інформації.

Завдання 10.3

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

1. Результати виконання функцій із заголовкового файлу **ModulesПрізвище.h** $s_calculation$ з аргументами x , y , z ;
2. Число b у двійковому коді.

Програмна реалізація

modulesbuliukin.h

```
#ifndef MODULESBULIUKIN_H
#define MODULESBULIUKIN_H

#include <iostream>
#include <fstream>
#include <time.h>
#include <math.h>
#include <windows.h>
```

```

void setUTF8();
void printDeveloperInfo();
float s_calculation(float x, float y, float z);

void task1();
void task2();
void task3(float x, float y, float z, int b);

#endif // MODULESBULIUKIN_H

```

modulesbuliukin.cpp

```

#include "modulesbuliukin.h"

using namespace std;

void setUTF8() {
    SetConsoleOutputCP(CP_UTF8);
    SetConsoleCP(CP_UTF8);
}

float s_calculation(float x, float y, float z) {
    if (x + y <= 0) {
        cout << "Помилка: x + y має бути більше 0.";
    }

    float S = sin(x) + (3 * pow(y, 5)) / sqrt(x + y);

    return S;
}

void task1() {
    string word;
    ifstream in("input.txt");
    if (!in.is_open()) {
        cout << "Помилка відкриття вхідного файлу!" << endl;
        return;
    }
    in >> word;
    in.close();

    ofstream out("output.txt");
    if (!out.is_open()) {
        cout << "Помилка відкриття вихідного файлу!" << endl;
        return;
    } else {
        cout << "Файл відкритий успішно!\n";
    }

    out << "Розробник: Булюкін В.Ю." << endl
        << "Центральноукраїнський Національний Технічний Університет" << endl
        << "Кропивницький, Україна" << endl
        << "Рік розробки: 2025" << endl << endl;

    int vowels = 0;
    string ukrainianVowels = "аеєііііоуяАЕЄІІІІОУЯ";
    for (char c : word) {
        if (ukrainianVowels.find(c) != string::npos) {
            vowels++;
        }
    }
}

```

```

    }
}
out << "Кількість голосних літер у слові \"" << word << "\": " << vowels << endl;

string poem = "До щастя не пускає лінощів орава. У чім воно ніхто не знає до пуття.
"
                "Навчитись радісно робити кожну справу Найперше правило щасливого
життя.";

    out << "Слово \"" << word << "\" " << (poem.find(word) != string::npos ? "е" :
"відсутнє")
        << " у вірші Віталія Іващенко" << endl;
    out.close();
}

void task2() {
    ifstream in("input.txt");
    if (!in.is_open()) {
        cout << "Помилка відкриття вхідного файлу!" << endl;
        return;
    }

    string content, line;
    while (getline(in, line)) {
        content += line + "\n";
    }
    in.close();

    int consonants = 0;
    for (char c : content) {
        if (isalpha(c) && !strchr("аеєіііоуюяАЕЄІІІОУЮЯ", c)) {
            consonants++;
        }
    }

    ofstream out("input.txt", ios::app);
    if (!out.is_open()) {
        cout << "Помилка відкриття файлу для дозапису!" << endl;
        return;
    }

    time_t now = time(0);
    char* dt = ctime(&now);

    out << "\nКількість приголосних літер у файлі: " << consonants << endl;
    out << "Дата та час дозапису: " << dt;
    out.close();
}

void task3(float x, float y, float z, int b) {
    ofstream out("output.txt", ios::app);
    if (!out.is_open()) {
        cout << "Помилка відкриття файлу для дозапису!" << endl;
        return;
    }

    out << "\nРезультати обчислень:" << endl;
    out << "s_calculation(" << x << ", " << y << ", " << z << ") = " <<
s_calculation(x, y, z) << endl;

```

```

out << "Число " << b << " у двійковому коді: ";
for (int i = 31; i >= 0; i--) {
    out << ((b >> i) & 1);
}
out << endl;

out.close();
}

```

TasksBuliukin (main.cpp)

```

#include "modulesbuliukin.h"

using namespace std;

int main()
{
    setUTF8();

    ifstream checkFile("input.txt");
    if (!checkFile.is_open()) {
        ofstream createFile("input.txt");
        if (createFile.is_open()) {
            createFile << "щастя" << endl;
            createFile.close();
            cout << "Створено файл input.txt з тестовим словом 'щастя'" << endl;
        } else {
            cout << "Помилка! Неможливо створити файл input.txt" << endl;
            return 1;
        }
    }
    checkFile.close();

    task1();
    task2();
    task3(1.5, 2.3, 3.7, 42);

    return 0;
}

```

TestDriver (main.cpp)

```

#include <iostream>
#include <fstream>
#include <cassert>
#include <cmath>
#include "modulesbuliukin.h"

using namespace std;

void test_s_calculation() {
    // Test case 1: normal values
    float result1 = s_calculation(2.0f, 3.0f, 1.0f);
    assert(abs(result1 - (sin(2.0f) + (3 * pow(3.0f, 5)) / sqrt(2.0f + 3.0f))) <
0.0001f);

    // Test case 2: different values
    float result2 = s_calculation(1.5f, 2.3f, 3.7f);
}

```

```

    assert(abs(result2 - (sin(1.5f) + (3 * pow(2.3f, 5)) / sqrt(1.5f + 2.3f))) <
0.0001f);

    cout << "s_calculation tests passed!" << endl;
}

void test_task1() {
    // Test input.txt creation and reading
    ofstream test_file("input.txt");
    test_file << "тест" << endl;
    test_file.close();

    ifstream check_file("input.txt");
    string content;
    check_file >> content;
    assert(content == "тест");
    check_file.close();

    // Test task1 output
    task1();
    ifstream output_file("output.txt");
    string line;
    bool found_developer = false;
    bool found_vowels = false;

    while (getline(output_file, line)) {
        if (line.find("Булюкін") != string::npos) found_developer = true;
        if (line.find("Кількість голосних") != string::npos) found_vowels = true;
    }
    output_file.close();

    assert(found_developer && found_vowels);
    cout << "File operations tests passed!" << endl;
}

void test_task2() {
    // Create test input
    ofstream test_file("input.txt");
    test_file << "абв" << endl;
    test_file.close();

    task2();

    // Check if consonants count was appended
    ifstream check_file("input.txt");
    string content((istreambuf_iterator<char>(check_file),
        istreambuf_iterator<char>()));
    check_file.close();

    assert(content.find("Кількість приголосних") != string::npos);
    assert(content.find("Дата та час") != string::npos);

    cout << "Task2 tests passed!" << endl;
}

void test_task3() {
    task3(1.5f, 2.3f, 3.7f, 42);

    ifstream output_file("output.txt");
    string content((istreambuf_iterator<char>(output_file),

```

```

        istreambuf_iterator<char>());
output_file.close();

assert(content.find("s_calculation(1.5, 2.3, 3.7)") != string::npos);
assert(content.find("Число 42 у двійковому коді") != string::npos);

cout << "Task3 tests passed!" << endl;
}

int main() {
    setUTF8();

    cout << "Starting tests..." << endl;

    test_s_calculation();
    test_task1();
    test_task2();
    test_task3();

    cout << "All tests passed successfully!" << endl;

    return 0;
}

```

Висновок

У ході виконання лабораторної роботи №10 з дисципліни "Базові методології та технології програмування" було реалізовано програмні модулі, що демонструють роботу з текстовими файлами та файловими потоками. Основні цілі роботи, такі як засвоєння навичок роботи з файловим введенням-виведенням у C++, обробка текстових даних (підрахунок голосних і приголосних літер, аналіз вмісту тексту), формування структурованих звітів, а також використання файлових потоків ofstream та ifstream, були успішно досягнуті.

Відповіді на контрольні запитання

1. **Яке призначення та синтаксис запису блоку-контроля try-throw-catch у мові програмування C++?** Блок try-throw-catch призначений для обробки виключних ситуацій (винятків). Конструкція дозволяє відловлювати та обробляти помилки. Синтаксис запису:

```

try {
    // код, який може викликати виняток
    if (умова) throw виняток;
} catch (тип_винятку параметр) {
    // обробка винятку
}

```

2. **Наведіть приклад опису й використання міжмодульної змінної.** Міжмодульна змінна оголошується з використанням ключового слова extern:

```

// У заголовковому файлі (header.h)
extern int глобальна_змінна;
// В одному з модулів (module1.cpp)
int глобальна_змінна = 10;
// В іншому модулі (module2.cpp)
#include "header.h"
void функція() {
    глобальна_змінна = 20; // доступ до змінної з іншого модуля
}

```

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

Об'єкти, описані в тілі функції main, мають локальну область видимості і доступні тільки всередині цієї функції. Вони створюються при вході в функцію і знищуються при виході з неї. Доступ до них можливий тільки з коду всередині функції main або із вкладених блоків.

4. Здійсніть порівняльний аналіз змінної типу enum та масиву.

1. enum - це перелічуваний тип, який дозволяє визначити набір іменованих констант:

- a. Займає фіксований розмір пам'яті (зазвичай розмір int)
- b. Значення є цілочисельними константами
- c. Не можна додавати нові значення під час виконання
- d. Використовується для зберігання фіксованого набору варіантів

2. Масив - це колекція елементів одного типу:

- a. Розмір пам'яті залежить від кількості та типу елементів
- b. Може містити будь-які значення відповідного типу
- c. Елементи доступні за індексами
- d. Дозволяє зберігати та обробляти набори даних
- e. Можна змінювати елементи під час виконання