

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 І/О

ВИКОНАЛА
студентка академічної групи
КБ-22-2

_____ Є. СПРИНЧАН

ПЕРЕВІРИВ
викладач кафедри кібербезпеки
та програмного забезпечення

_____ Олександр СОБІНОВ

ТЕМА: Реалізація програмних модулів оброблення даних складових типів з файловим I/O

МЕТА: набуття ґрунтовних вмінь і практичних навичок реалізації у Code::Blocks IDE мовою 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 (проект ModulesПрізвище лабораторних робіт №8–9);
2. Реалізувати тестовий драйвер автоматизованої перевірки програмних модулів розв'язування задач 10.1–10.3.

ВАРІАНТ 10

— ВХІДНИЙ ТЕКСТ - ВМІСТ ВХІДНОГО ТЕКСТОВОГО ФАЙЛУ —

Як парость виноградної лози, плекайте мову.
Пильно й ненастанно політь бур'ян.
Чистіша від сльози вона хай буде.
Вірно і слухняно нехай вона щоразу служить вам,
Хоч і живе своїм живим життям.

Рисунок 1. - Текстовий файл input.txt

Завдання 1

Задача 10.1:

— ЗАДАЧА 10.1 —

У вихідний текстовий файл записати:

- авторську інформацію: ім'я й прізвище розробника модуля, установа/організація, місто, країна, рік розробки;
- випадкове число від 10 до 100;
- повідомлення, чи текст віршу із вхідного файла має пунктуаційні помилки (відповідно до авторського оригіналу).

Рисунок 2. - Задача 10.1

Задача 10.2:*— ЗАДАЧА 10.2 —*

У *вхідний* текстовий файл дописати:

- дату дозапису інформації у форматі «ДД.ММ.РРРР».

Рисунок 3. - Задача 10.2

Задача 10.3:*— 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.

Рисунок 4. - Задача 10.3

Лістинг ModulesSprynchan:

```
#include <math.h>
#include <iostream>
#include <vector>
#include <algorithm>
#include <fstream>
#include <cstdlib>
#include <ctime>
#include <string>
#include "ModulesSprynchan.h"

double s_calculation(double x, double y, double z) {
    double S = z * sin(pow(x, 2) * y) + sqrt(fabs(z - 12 * x)) / pow(y, 3);
    return S;
}
```

```
double el_calculation(int cons_energy) {  
    double price;  
  
    if (cons_energy <= 150) {  
        price = 0.3084;  
    } else if (cons_energy <= 800) {  
        price = 0.4194;  
    } else {  
        price = 1.3404;  
    }  
  
    return cons_energy * price;  
}
```

```
int Beaufort_scale(double speed) {  
    if (speed < 0.3) return 0;  
    if (speed <= 1.5) return 1;  
    if (speed <= 3.3) return 2;  
    if (speed <= 5.4) return 3;  
    if (speed <= 7.9) return 4;  
    if (speed <= 10.7) return 5;  
    if (speed <= 13.8) return 6;  
    if (speed <= 17.1) return 7;  
    if (speed <= 20.7) return 8;  
    if (speed <= 24.4) return 9;  
    if (speed <= 28.4) return 10;  
    if (speed <= 32.6) return 11;  
    return 12;  
}
```

```

int count_zeros_or_ones(int n) {
    int bit = (n >> 9) & 1;
    int count = 0;
    if (bit == 1) {
        while (n) {
            count += (n & 1) ? 0 : 1;
            n >>= 1;
        }
    } else {
        while (n) {
            count += (n & 1) ? 1 : 0;
            n >>= 1;
        }
    }
    return count;
}

```

```

bool checkPunctuation(string text) {
    string originalText = "Як парость виноградної лози, плекайте мову.\n"
        "Пильно й ненастанно політь бур'ян.\n"
        "Чистіта від сльози вона хай буде.\n"
        "Вірно і слухняно нехай вона щоразу служить вам,\n"
        "Хоч і живе своїм живим життям.";

    for (int i = 0; i < text.length(); i++) {
        if (ispunct(text[i]) || isspace(text[i])) {
            text.erase(i--, 1);
        }
    }
}

```

```

    return text == originalText;
}

void append_date(const std::string& filename) {
    std::ofstream outfile(filename, std::ios_base::app);
    if (!outfile) {
        std::cerr << "Помилка: не вдалося відкрити файл для дозапису!\n";
        return;
    }

    std::time_t now = std::time(nullptr);
    std::tm* now_tm = std::localtime(&now);
    char date[11];
    std::strftime(date, sizeof(date), "%d.%m.%Y", now_tm);

    outfile << "\nДата дозапису: " << date << "\n";
    outfile.close();
}

void myFunction() {
    double x, y, z;
    int b;

    cout << "Введіть числа x, y, z та натуральне число b:" << endl;
    cin >> x >> y >> z >> b;

    ofstream outFile;
    outFile.open("output.txt", ios_base::app);

```

```

if (!outFile) {
    cerr << "Не вдалося відкрити файл output.txt" << endl;
    return;
}

outFile << "Результати функції s_calculation з аргументами " << x << ", " << y <<
", " << z << ": " << s_calculation(x, y, z) << endl;

int binary_b[32];
int i = 0;
while (b > 0) {
    binary_b[i] = b % 2;
    b /= 2;
    i++;
}
outFile << "Число " << b << " у двійковому коді: ";
for (int j = i - 1; j >= 0; j--) {
    outFile << binary_b[j];
}
outFile << endl;

outFile.close();
}

```

Лістинг TestDriver:

```

#include "ModulesSprynchan.h"
#include <iostream>
#include <fstream>
#include <cstdlib>
#include <ctime>

```



```

}

std::time_t now = std::time(nullptr);
std::tm* now_tm = std::localtime(&now);
char date[11];
std::strftime(date, sizeof(date), "%d.%m.%Y", now_tm);

outfile << "\n????? : " << date << "\n";
outfile.close();

double x, y, z;
int b;

cout << "????? x, y, z ? ? ? ? ? ? ? ? ? ? b:" <<
endl;
cin >> x >> y >> z >> b;

ofstream outFile;
outFile.open("output.txt", ios_base::app);

if (!outFile) {
    cerr << "????? output.txt" << endl;
    return;
}

outFile << "????? s_calculation ? ? ? ? ? ? ? ? ? ?
" << x << ", " << y << ", " << z << ": " << s_calculation(x, y, z) << endl;

int binary_b[32];
int i = 0;

```

```

while (b > 0) {
    binary_b[i] = b % 2;
    b /= 2;
    i++;
}
outFile << "00000 " << b << " 00000000000000000000: ";
for (int j = i - 1; j >= 0; j--) {
    outFile << binary_b[j];
}
outFile << endl;
}
outFile.close();

return 0;
}

```

ВИСНОВКИ

Під час підготовки до виконання лабораторної роботи було належно опрацьовано рекомендовану літературу та контрольні запитання. Після ознайомлення з порядком проведення лабораторної роботи, було розпочато її виконання.

В ході виконання завдань та написання лістингу до завдань, проблем не було виявлено. Робота над цими завданням була плідна, зрозуміла та цікава. До звіту з лабораторної роботи додані лістинги до кожного завдання, алгоритми та тест-сьюти.

Завдання лабораторної роботи були цікаві та корисні у майбутньому. Я набуття ґрунтовних вмінь і практичних навичок реалізації метода функціональної декомпозиції задач, метода модульного (блочного) тестування, представлення мовою програмування C/C++ даних скалярних типів, арифметичних і логічних операцій, потокового введення й виведення інформації, розроблення програмних модулів та засобів у кросплатформовому середовищі Code::Blocks (GNU GCC Compiler).

Мета цієї 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 символьної інформації. Для виконання даної лабораторної роботи потрібно приблизно - 4 академічні

години.

Також потрібно таке обладнання:

- ~ персональний комп'ютер з операційною системою Windows; v вільне кросплатформове Code::Blocks IDE (www.codeblocks.org); v текстовий редактор (OpenOffice Writer, Microsoft Word або ін.);
- v - файл-шаблон тестового набору Artifact_TEST_SUITE_lab.doc;
- v власний обліковий запис на GitHub

При підготовці до лабораторної роботи я отримала такі завдання.

- Реалізувати програмні модулі розв'язування задач 10.1-10.3 як складові статичної бібліотеки libModules.a (проектModules лабораторних робіт №8-9).
- Реалізувати тестовий драйвер автоматизованої перевірки програмних модулів розв'язування задач 10.1-10.3.

Я дотримувалась такого плану виконання лабораторної роботи №9:

- 1) Спочатку я зайшла на мій Git Repositories та отримав завдання для виконання лабораторної роботи №10.
- 2) Завантажила Git Repositories на мій диск.
- 3) Змінила вміст файлу README md, вказавши: тему, мету, варіант та завдання 10.1-10.3.
- 4) Створила теки: prj, SoftWare, TestSuite, Report. Також завантажив все на репозиторій.
- 5) Здійснила аналіз задач 10.1 - 10.3.
- 6) Задokumentувала артефакти аналізу та постановки задачі в звіті.

- 7) З лабораторної роботи No9, перемістила статичну бібліотеку ModulesKaras в папку prj.
- 8) Реалізувала функції: Task_10_1, Task_10_2, Task_10_3.
- 9) Скомпілювала проєкт статичної бібліотеки.
- 10) З лабораторної роботи No9, перемістила заголовковий файл ModulesKaras.h, в ньому вписала прототипи функцій Task_10_1, Task_10_2, Task_10_3.
- 11) В Code::Blocks створила проєкт консольного застосунку C++, іменував його як TestDriver.
- 12) Реалізувала мовою програмування C++, тестовий драйвер, підключила статичну бібліотеку та заголовковий файл в налаштуваннях компілятора.
- 13) Скопіювала текст з консольного вікна та перемістила його в ЗВІТ.
- 14) Закінчила працювати з звітом.
- 15) Надіслала всі файли на Git Repositorie.

Данна лабораторна робота розширила мої знання.

ДОДАТОК А

Назва тестового набору Test Suite Description	UnitTesting10.1 - 10.3
Назва проєкта / ПЗ Name of Project / Software	Unit_Testing10.1 - 10.3
Рівень тестування Level of Testing	модульний / Unit Testing
Автор тест-сьюта Test Suite Author	Єлизавета Спринчан
Виконавець Implementer	Єлизавета Спринчан

Artifact: Test Suite Date:

12.05.2023

10.1			
Preliminary Steps	Action	Expected Result	Result
<p>Вихідний файл output.txt:</p> <p>Вхідний файл input.txt: Як парость виноградної лози, плекайте мову. Пильно й ненастанно політь бур'ян. Чистіша від сльози вона хай буде. Вірно і слухняно нехай вона щоразу служить вам, Хоч і живе своїм живим життям.</p>	<p>Викликаємо функцію checkPunctuation</p>	<p>Вихідний файл output.txt: Спринчан Єлизавета м.Кропивницький 12.05.2023 (число від 10 до 100) Пунктуаційних помилок немає</p> <p>Вхідний файл input.txt: Як парость виноградної лози, плекайте мову. Пильно й ненастанно політь бур'ян. Чистіша від сльози вона хай буде. Вірно і слухняно нехай вона щоразу служить вам, Хоч і живе своїм живим життям.</p>	<p>Passed</p>

10.2			
Preliminary Steps	Action	Expected Result	Result
<p>Вихідний файл input.txt: <i>Як парость виноградної лози, плекайте мову. Пильно й ненастанно політь бур'ян. Чистіша від сльози вона хай буде. Вірно і слухняно нехай вона щоразу служить вам, Хоч і живе своїм живим життям.</i></p>	<p>Викликаємо функцію append_date</p>	<p>Вихідний файл input.txt: <i>Як парость виноградної лози, плекайте мову. Пильно й ненастанно політь бур'ян. Чистіша від сльози вона хай буде. Вірно і слухняно нехай вона щоразу служить вам, Хоч і живе своїм живим життям.</i></p> <p>12.05.2023</p>	<p>Passed</p>

[illegible]

