

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА КІБЕРБЕЗПЕКИ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ЗВІТ

з лабораторної роботи № 10

з дисципліни

«Базові методології та технології програмування»

на тему:

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

Виконав:

студент академічної групи КН-22

ПІВНЕНКО О.М.

Перевірив:

викладач

СОБІНОВ О.Г.

Лабораторна робота № 10

Варіант № 15

ТЕМА: Реалізація програмних модулів оброблення даних складових типів з файловим введенням/виведенням.

МЕТА: набуття 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.

ЗАВДАННЯ:

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.

10.2. У вхідний текстовий файл дописати:

- Перелік Державних символів України згідно зі ст.20 Конституції України.
- Дату дозаписування інформації.

10.3. Вхідні дані – числові значення x , y , z та натуральне число b . У вхідний текстовий файл дописати:

- Результати виконання функцій із заголовкового файлу `ModulesПрізвище.h s_calculation` з аргументами x , y , z ;
- Число b у двійковому коді.

ХІД ЛАБОРАТОРНОЇ РОБОТИ

Вихідний код проектів ModulesPivnenko та TestDriver зазначений у додатках 1 та 2. Тест-сюїти до завдань зазначені у додатках 3, 4, 5. Результати тестування (текст з консолі) зазначені у додатку 6.

Завдання 10.1.

Вихідний код функції, що реалізує завдання, зазначений у лістингу 10.1.

Лістинг 10.1.

```
bool Task_10_1(string inputPath){
    string toWrite = "Розробник: О. Півненко.\nCНТУ, м.Кропивницький, Україна, 2023 р.";
    string readLine;
    string text;
    string toFind[4] = {"програма", "модуль", "студент", "програміст"};

    string outputPath = "outfile.txt";
    ofstream outStream;
    ifstream readFile;

    readFile.open(inputPath, ios_base::in);
    outStream.open("outfile.txt", ios_base::out);

    try
    {
        if(!outStream.is_open() || !readFile.is_open())
            throw exception();

        while(!readFile.eof()){
            getline(readFile, readLine);
            text += readLine;
            text += "\n";
        }
        readFile.close();

        outStream << toWrite << endl;
        outStream << "К-ть символів у тексті: " << text.length() << endl;
        outStream << "Знайдені слова у тексті: " << findWords(text, toFind, 4);

        outStream.close();
    }
    catch(...){
        cout << "Файл не можна відкрити." << endl;
        return false;
    }

    return true;
}
```

Завдання 10.2.

Вихідний код функції, що реалізує завдання 10.2., зазначений у лістингу 10.2. Варто також зауважити, що для реалізації завдання було використано функції із заголовкового файлу `ctime`.

Лістинг 10.2.

```
bool Task_10_2(string inputPath){
    try{
        string toWrite = "Державними символами України є Державний Прапор України,
Державний Герб України і Державний гімн України.\n";

        time_t now = time(nullptr);
        tm *local_time = localtime(&now);

        ofstream writer;
        writer.open(inputPath, ios_base::out | ios_base::app);

        if(!writer.is_open())
            throw exception();

        writer << toWrite << endl;
        writer << "Час запису: " << local_time -> tm_hour << ":" << local_time ->
tm_min << ":" << local_time -> tm_sec << endl;
        writer << "Дата запису: " << local_time -> tm_mday << "." << local_time ->
tm_mon + 1 << "." << local_time -> tm_year + 1900 << endl;

        writer.close();
    }
    catch(...){
        cout << "Файл не можна відкрити." << endl;
        return false;
    }
    return true;
}
```

Завдання 10.3.

Вихідний код функції, що реалізує завдання 10.3., зазначений у лістингу 10.3. Слід зауважити, що тут також використовується функція toBinary в якості допоміжної, тому вона наведена у лістингу 10.4.

Лістинг 10.3.

```
bool Task_10_3(int x, int y, int z, unsigned short b, string inputPath){
    try{
        ofstream writer;
        writer.open(inputPath, ios_base::out | ios_base::app);

        if(!writer.is_open())
            throw exception();

        float result_s = s_calculation(x, y, z);

        writer << "Результати обчислення s_calculation: " << result_s << endl;
        writer << "Число b у двійковому коді: " << toBinary(b) << endl;

        writer.close();
    }
    catch(...){
        cout << "Файл не можна відкрити.";
        return false;
    }

    return true;
}
```

Лістинг 10.4.

```

string toBinary(unsigned short b){
    int cnt = sizeof(b);
    unsigned short mask = 32768;
    string binary_view = "";

    for (int i = 0; i < (cnt * 8); i++)
    {
        int temp_val = b & mask;
        if(temp_val == 0)
            binary_view += "0";
        else
            binary_view += "1";
        mask = mask >> 1;
    }

    return binary_view;
}

```

ВИСНОВОК

У ході виконання лабораторної роботи я підвищив свій рівень знань із мови програмування C++. Попрацював із текстовими редакторами Sublime Text, Visual Studio Code, Notepad++, IDE Code::Blocks. Реалізував функції, метою яких було вирішення завдань 10.1, 10.2, 10.3.. Склав тест-сьюти, на основі яких зробив модульне тестування. Реалізував тест-драйвер, за допомогою якого реалізував unit-тестування. Виявив помилки, які були потім усунуті (наприклад, неправильне конвертування числа `b` у двійковий вид), а також поліпшив загальний вихідний код модуля. Попрацював із заголовковим файлом `<ctime>`, за допомогою якого була реалізована фіксація часу та дати запису. Закріпив свої знання щодо файлового вводу/виводу, використавши класи `ofstream`, `istream`. Ознайомився із стандартними методами цих класів. Згадав основи роботи десяткової та двійкової систем числення, конвертацію чисел із однієї системи в іншу. Варто також зауважити, що під час виконання роботи я попрацював ще й із вказівниками (під час роботи із `<ctime>`), а також закріпив свої знання щодо керування пам'яттю. Не можна також перелічити численну к-ть малих проблем, із якими зіштовхнувся під час реалізації модуля. Усе це неодмінно підвищило мій рівень знань.

```

#include <math.h>
#include <iostream>
#include <iomanip>
#include <windows.h>
#include <fstream>
#include <ctime>
#include <ctype.h>
#define PI 3.1415

using namespace std;

string toBinary(unsigned short b){
    int cnt = sizeof(b);
    unsigned short mask = 32768;
    string binary_view = "";

    for (int i = 0; i < (cnt * 8); i++)
    {
        int temp_val = b & mask;
        if(temp_val == 0)
            binary_view += "0";
        else
            binary_view += "1";
        mask = mask >> 1;
    }

    return binary_view;
}

string findWords(string text, string words[], int cntWords){
    bool IsFind = false;
    int i = 0;
    int j = 0;
    int letter = 0;
    int text_size = text.length();
    string output = "";

    while(i < cntWords){
        while(j < text_size){
            if(words[i][letter] == text[j]){
                IsFind = true;
                letter += 1;
            }
            else{
                IsFind = false;
                letter = 0;
            }

            if(IsFind == true && letter == int(words[i].length()) && (text[j + 1] == ' ' ||
text[j + 1] == ':' || text[j + 1] == ';' || text[j + 1] == ',' || text[j + 1] == '!' ||
text[j + 1] == '?' || text[j + 1] == '-' || text[j + 1] == '.')){
                output += words[i] + " ";
                letter = 0;
                IsFind = false;
                break;
            }
            else if(IsFind == true && letter == int(words[i].length())){
                letter = 0;
                IsFind = false;
            }

            j += 1;
        }
        i += 1;
    }
}

```

```

        j = 0;
        i += 1;
    }

    return output;
}

float s_calculation(int x, int y, int z){
    float pow_res = pow(2*z + 1, x);

    float sqrt_res = sqrt(abs(y - ((float)1/(float)2) * z));

    float S = pow_res - sqrt_res + z + PI;

    return S;
}

bool Task_10_1(string inputPath){
    string toWrite = "Розробник: О. Півненко.\nЦНТУ, м.Кропивницький, Україна, 2023 р.";
    string readLine;
    string text;
    string toFind[4] = {"програма", "модуль", "студент", "програміст"};

    string outputPath = "outfile.txt";
    ofstream outStream;
    ifstream readFile;

    readFile.open(inputPath, ios_base::in);
    outStream.open("outputfile.txt", ios_base::out);

    try
    {
        if(!outStream.is_open() || !readFile.is_open())
            throw exception();

        while(!readFile.eof()){
            getline(readFile, readLine);
            text += readLine;
            text += "\n";
        }
        readFile.close();

        outStream << toWrite << endl;
        outStream << "К-ть символів у тексті: " << text.length() << endl;
        outStream << "Знайдені слова у тексті: " << findWords(text, toFind, 4);

        outStream.close();
    }
    catch(...){
        cout << "Файл не можна відкрити." << endl;
        return false;
    }

    return true;
}

bool Task_10_2(string inputPath){
    try{
        string toWrite = "Державними символами України є Державний Прапор України, Державний  
Герб України і Державний гімн України.\n";

        time_t now = time(nullptr);
        tm *local_time = localtime(&now);
    }

```

```

    ofstream writer;
    writer.open(inputPath, ios_base::out | ios_base::app);

    if(!writer.is_open())
        throw exception();

    writer << toWrite << endl;
    writer << "Час запису: " << local_time -> tm_hour << ":" << local_time -> tm_min <<
":" << local_time -> tm_sec << endl;
    writer << "Дата запису: " << local_time -> tm_mday << "." << local_time -> tm_mon
+ 1 << "." << local_time -> tm_year + 1900 << endl;

    writer.close();
}
catch(...){
    cout << "Файл не можна відкрити." << endl;
    return false;
}
return true;
}

bool Task_10_3(int x, int y, int z, unsigned short b, string inputPath){
    try{
        ofstream writer;
        writer.open(inputPath, ios_base::out | ios_base::app);

        if(!writer.is_open())
            throw exception();

        float result_s = s_calculation(x, y, z);

        writer << "Результати обчислення s_calculation: " << result_s << endl;
        writer << "Число b у двійковому коді: " << toBinary(b) << endl;

        writer.close();
    }
    catch(...){
        cout << "Файл не можна відкрити.";
        return false;
    }

    return true;
}

```



```

#include <iostream>
#include <iomanip>
#include <locale.h>
#include <windows.h>
#include <cmath>
#include "ModulesPivnenko.hpp"
#include <fstream>

using namespace std;

bool TestCase_10_3(int x, int y, int z, unsigned short b, string b_code, bool exceptRes,
string inPath){
    bool result = Task_10_3(x, y, z, b, inPath);

    if(result == exceptRes && b_code == toBinary(b))
        return true;
    else
        return false;
}

bool TestCase_10_2(bool expectRes, string inPath){
    bool result = Task_10_2(inPath);

    if(expectRes == result)
        return true;
    else
        return false;

    return false;
}

bool TestCase_10_1(bool expectRes, string inPath){
    bool result = Task_10_1(inPath);

    if(expectRes == result)
        return true;

    return false;
}

bool TestSuite_10_1(){
    const bool exp_results[7] = { true, false, true, false, true, false, true};
    const string paths[7] = { "new_f.txt", "newf.txt", "new_f.txt", " ", "new_f.txt",
"what_is_it", "new_f.txt"};
    int cnt_bools = 0;

    for(int i = 1; i <= 7; i += 1){
        bool case_res = TestCase_10_1(exp_results[i - 1], paths[i - 1]);

        cout << "Текст-кейс 10.1 за номером " << i;

        if(case_res == true){
            cout << " PASSED" << endl;
            cnt_bools += 1;
        }
        else{
            cout << " FAILED" << endl;
        }
        getchar();
    }

    if(cnt_bools == 7)

```

```

        return true;
    else
        return false;
}

bool TestSuite_10_2(){
    const string paths[7] = {"ukr.txt", "second_ukr.txt", "ukr.txt", "second_ukr.txt", " ",
"ukr.txt", " "};
    const bool expectRes[7] = {true, true, true, true, false, true, false};
    int cnt_bools = 0;

    for(int i = 1; i <= 7; i += 1){
        bool case_res = TestCase_10_2(expectRes[i - 1], paths[i - 1]);

        cout << "Текст-кейс 10.2 за номером " << i;

        if(case_res == true){
            cout << " PASSED" << endl;
            cnt_bools += 1;
        }
        else{
            cout << " FAILED" << endl;
        }

        getchar();
    }
    if(cnt_bools == 7)
        return true;
    else
        return false;
}

bool TestSuite_10_3(){
    const bool exceptRes[6] = {true, true, true, true, true, true};
    const string b_codes[6] = {"0000000001100100", "0000000001100100", "0000000000000100",
"0000010000000000", "0000001111111111", "0000101110111000"};
    const string paths[6] = {"main_f.txt", "main_f.txt", "main_f.txt", "main_f.txt",
"main_f.txt", "main_f.txt"};
    const unsigned short b_vals[6] = {100, 100, 4, 1024, 1023, 3000};
    const int x[6] = {3, 3, 1, 0, 0, 4};
    const int y[6] = {3, 3, 5, 0, 0, 7};
    const int z[6] = {3, 3, 2, 0, 0, 7};

    int cnt_bools = 0;

    for(int i = 1; i <= 6; i += 1){
        bool case_res = TestCase_10_3(x[i - 1], y[i - 1], z[i - 1], b_vals[i - 1], b_codes[i
- 1], exceptRes[i - 1], paths[i - 1]);
        int cnt_bools = 0;

        cout << "Текст-кейс 10.3 за номером " << i;

        if(case_res == true){
            cout << " PASSED" << endl;
            cnt_bools += 1;
        }
        else{
            cout << " FAILED" << endl;
        }

        getchar();
    }

    if(cnt_bools == 6)
        return true;
}

```

```
        else
            return false;
    }

int main(){
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    TestSuite_10_1();
    TestSuite_10_2();
    TestSuite_10_3();

    cout << "\nНатисніть ENTER щоб вийти із програми." << endl;
    getchar();
    return 0;
}
```

Додаток 3.

Artifact: Test Suite

Date: 4/29/2023

Назва тестового набору Test Suite Description	TS_lab10_1
Назва проекту / ПЗ Name of Project / Software	ModulesPivnenko
Рівень тестування Level of Testing	модульний / Module Testing
Автор тест-сьюта Test Suite Author	Олександр Півненко
Виконавець Implementer	Олександр Півненко

Ід-р тест- кейса / Test Case ID	Попередні кроки / Preliminary Steps	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування / Test Result
TC_01	Створити файл із назвою «new_f.txt». Заповнити його текстом «Привіт, програма!»	Викликати функцію Task_10_1, передати у якості аргумента inputPath = "new_f.txt".	У файл outputfile.txt записано О. Півненко, ЦНТУ, м.Кропивницький, 2023 р.	PASSED
TC_02	Створити файл із назвою «new_f.txt». Заповнити його текстом «Привіт, програма!»	Викликати функцію Task_10_1, передати у якості аргумента inputPath = "newf.txt".	Надпис у консолі: «Файл не можна відкрити.»	PASSED
TC_03	Створити файл із назвою «new_f.txt». Заповнити його текстом «Привіт, програма!»	Викликати функцію Task_10_1, передати у якості аргумента inputPath = "new_f.txt".	У файл outputfile.txt записано інформацію про розробника, к-ть символів (18), знайдені слова: програма.	PASSED
TC_04	Створити файл із назвою «new_f.txt». Заповнити його текстом «Привіт, програма!»	Викликати функцію Task_10_1, передати у якості аргумента inputPath = " ".	У вихідному файлі outputfile.txt не записано нічого.	PASSED
TC_05	Створити файл із назвою «new_f.txt». Заповнити його текстом «програма, зроблена студентом. модуль також».	Викликати функцію Task_10_1, передати у якості аргумента inputPath = "new_f.txt".	У вихідному файлі записано інформація про розробника, к-ть символів (44), знайдені слова: модуль, програма	PASSED
TC_06	Створити файл із назвою «new_f.txt». Заповнити його текстом «програмулька, зроблена студентом.».	Викликати функцію Task_10_1, передати у якості аргумента inputPath = "what_is_it".	Надпис у консолі: «Файл не можна відкрити.»	PASSED
TC_07	Створити файл із назвою «new_f.txt». Заповнити його текстом «програма, студент, модуль, модуль, програміст».	Викликати функцію Task_10_1, передати у якості аргумента inputPath = "new_f.txt".	У вихідному файлі записано інформацію про розробника, к-ть символів (47), знайдені слова: програма, модуль, студент, програміст.	PASSED

Додаток 4.

Artifact: Test Suite

Date: 4/29/2023

Назва тестового набору Test Suite Description	TS_lab10_2
Назва проекту / ПЗ Name of Project / Software	ModulesPivnenko
Рівень тестування Level of Testing	модульний / Module Testing
Автор тест-сюїта Test Suite Author	Олександр Півненко
Виконавець Implementer	Олександр Півненко

Ід-р тест- кейса / Test Case ID	Попередні кроки / Preliminary Steps	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування / Test Result
TC_01	Створити файл «ukr.txt». Нічого не записувати: залишити пустим.	Викликати Task_10_2 та передати у якості аргумента «ukr.txt»	У файл «ukr.txt» записано список державних символів України та дату/час.	PASSED
TC_02	Створити файл «ukr.txt». Нічого не записувати: залишити пустим.	Викликати Task_10_2 та передати у якості аргумента «second_ukr.txt»	Файл «second_ukr.txt» було створено, в нього записано дату й час та список державних символів.	PASSED
TC_03	Створити файл «ukr.txt». Нічого не записувати: залишити пустим. Змінити на ПК дату на 1993 р.	Викликати Task_10_2 та передати у якості аргумента «ukr.txt»	У файл «ukr.txt» записано дату й час та список державних символів. Але у даті зазначено 1993 р.	PASSED
TC_04	Створити файл «ukr.txt». Нічого не записувати: залишити пустим. Змінити на ПК дату на 1993 р.	Викликати Task_10_2 та передати у якості аргумента «second_ukr.txt»	Файл «second_ukr.txt» було створено, в нього записано дату й час та список державних символів. Але у даті зазначено 1993 р.	PASSED
TC_05	Створити файл «ukr.txt». Нічого не записувати: залишити пустим.	Викликати Task_10_2 та передати у якості аргумента « »	Файл не було створено, натомість у консолі зазначено «Файл не можна відкрити.»	PASSED
TC_06	Створити файл «ukr.txt». Нічого не записувати: залишити пустим. Змінити на ПК дату на 2000 р., а час на довільний.	Викликати Task_10_2 та передати у якості аргумента «ukr.txt»	Файл «second_ukr.txt» було створено, в нього записано дату 2000р, а час довільний (в залежності від вибору).	PASSED
TC_07	Створити файл «ukr.txt». Нічого не записувати: залишити пустим. Змінити на ПК дату на 2000 р., а час на довільний.	Викликати Task_10_2 та передати у якості аргумента « »	Файл не було створено, натомість у консолі зазначено «Файл не можна відкрити.»	PASSED

Додаток 5.

Artifact: Test Suite

Date: 4/29/2023

Назва тестового набору Test Suite Description	TS_lab10_3
Назва проекту / ПЗ Name of Project / Software	ModulesPivnenko
Рівень тестування Level of Testing	модульний / Module Testing
Автор тест-сюїта Test Suite Author	Олександр Півненко
Виконавець Implementer	Олександр Півненко

Ід-р тест-кейса / Test Case ID	Попередні кроки / Preliminary Steps	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування / Test Result
TC_01	Створити файл «main_f.txt», нічого туди не записувати.	Викликати функцію Task_10_3 із аргументами (3, 3, 3, 100, "main_f.txt")	У файл main_f.txt записано результат s_calculation (347.917), а також b у двійковому коді (0000000001100100)	PASSED
TC_02	Не створювати жодного файла. Видалити main_f.txt.	Викликати функцію Task_10_3 із аргументами (3, 3, 3, 100, "main_f.txt")	Файл main_f.txt буде створено, результат s_calculation (347.917), а також b у двійковому коді (0000000001100100)	PASSED
TC_03	Не створювати жодного файла. Видалити main_f.txt.	Викликати функцію Task_10_3 із аргументами (1, 5, 2, 4, "main_f.txt")	Файл main_f.txt буде створено, записано результат обчислення s_calculation (8.1415), а також b у двійковому коді (0000000000000100)	PASSED
TC_04	Не створювати жодного файла. Видалити main_f.txt.	Викликати функцію Task_10_3 із аргументами (0, 0, 0, 1024, "main_f.txt")	Файл main_f.txt буде створено, записано результат обчислення s_calculation (4,1415), а також b у двійковому коді (0000010000000000)	PASSED
TC_05	Створити файл «next_f.txt». Нічого не записувати у нього. Видалити main_f.txt.	Викликати функцію Task_10_3 із аргументами (0, 0, 0, 1023, "main_f.txt")	Файл main_f.txt буде створено, записано результат s_calculation (4,1415), а також b у двійковому коді (0000001111111111). Файл «next_f.txt» залишиться без змін.	PASSED
TC_06	Створити файл «next_f.txt». Записати у нього будь-які слова кирилицею. Видалити main_f.txt.	Викликати функцію Task_10_3 із аргументами (6, 7, 7, 3000, "main_f.txt")	Файл main_f.txt буде створено, записано результат s_calculation (50633.3), а також b у двійковому коді (0000101110111000). Файл «next_f.txt» залишиться без змін.	PASSED

Текст-кейс 10.1 за номером 1 PASSED

Файл не можна відкрити.

Текст-кейс 10.1 за номером 2 PASSED

Текст-кейс 10.1 за номером 3 PASSED

Файл не можна відкрити.

Текст-кейс 10.1 за номером 4 PASSED

Текст-кейс 10.1 за номером 5 PASSED

Файл не можна відкрити.

Текст-кейс 10.1 за номером 6 PASSED

Текст-кейс 10.1 за номером 7 PASSED

Текст-кейс 10.2 за номером 1 PASSED

Текст-кейс 10.2 за номером 2 PASSED

Текст-кейс 10.2 за номером 3 PASSED

Текст-кейс 10.2 за номером 4 PASSED

Файл не можна відкрити.

Текст-кейс 10.2 за номером 5 PASSED

Текст-кейс 10.2 за номером 6 PASSED

Файл не можна відкрити.

Текст-кейс 10.2 за номером 7 PASSED

Текст-кейс 10.3 за номером 1 PASSED

Текст-кейс 10.3 за номером 2 PASSED

Текст-кейс 10.3 за номером 3 PASSED

Текст-кейс 10.3 за номером 4 PASSED

Текст-кейс 10.3 за номером 5 PASSED

Текст-кейс 10.3 за номером 6 PASSED

Натисніть ENTER щоб вийти із програми.