

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 введенням/виведенням

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

ВИКОНАВ
студент академічної групи КІ-22-2
Ясініцький Я. О.

ПЕРЕВІРИВ
ст. викладач кафедри кібербезпеки
та програмного забезпечення
Собінов О.Г.

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

Мета: Мета 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 символічної інформації.

Завдання: 1. Реалізувати програмне забезпечення розв'язування задачі 10.1.
2. Реалізувати програмне забезпечення розв'язування задачі 10.2.
3. Реалізувати програмне забезпечення розв'язування задачі 10.3.

Варіант 8

Лістинг вихідного коду із ModulesYasinitskiy

```
#include <cmath>
#include <ctime>
#include <string>
#include <fstream>
#include <codecvt>
#include <bitset>
#include <limits>
#include "ModulesYasinitskiy.h"

using namespace std;

wstring letters = L"абвгдеєжзиййклмнопрстуфхцчшщьюя";
wstring replacedLetters = L"ктулй";

float s_calculation(int x, int y, int z) {
    if (x > 3)
        return pow(3*sin(sqrt(12*x+log10(x-3))), y)+(z/(float)x);
    else
        return NAN;
}

Bonus getBonus(float value) {
    Bonus bonus;
    bonus.bonusBalance = 0;

    if (value >= 100 && value < 200)
        bonus.bonusBalance = 1;
    else if (value >= 200 && value < 500)
        bonus.bonusBalance = 5;
    else if (value >= 500 && value < 1000)
```

```

        bonus.bonusBalance = 10;
    else if (value >= 1000 && value < 2500)
        bonus.bonusBalance = 50;
    else if (value >= 2500 && value < 5000)
        bonus.bonusBalance = 150;
    else if (value >= 5000)
        bonus.bonusBalance = 300;

    bonus.discountedSum = value - bonus.bonusBalance * 0.25;
    return bonus;
}

```

```

float getCelsius(float fahrenheit) {
    return (5.0/9)*(fahrenheit-32);
}

```

```

int t9_3(int number) {
    unsigned int count = 0;
    if (number > 0 || number <= 51950) {
        bool set = (number >> 4) & 1;
        while (number) {
            count += (number & 1) == set;
            number >>= 1;
        }
    } else {
        return -1;
    }
    return count;
}

```

```

int t10_1(string inputFile, string outputFile) {
    wstring line;
    int number = 0;
    wifstream indata;
    indata.open(inputFile);
    wofstream outdata;
    outdata.open(outputFile);

    if (!indata || !outdata)
        return 1;

    indata.imbue(locale(locale(), new codecvt_utf8<wchar_t>));
    outdata.imbue(locale(locale(), new codecvt_utf8<wchar_t>));
}

```

```

    outdata << L"Розробник: Yasinitskiy Yan" << endl << L"Установа/організація:
Центральноукраїнський національний технічний університет" << endl << L"Місто:
Кропивницький" << endl << L"Країна: Україна" << endl << L"Рік розробки: 2023" <<
endl << endl;

```

```

outdata << L"Речення: ";

while (getline(indata,line)) {
    for (unsigned int i = 0; i < line.length(); i++) {
        for (unsigned int j = 0; j < letters.length(); j++)
        {
            if (tolower(line[i]) == letters.at(j))
            {
                number += 1;
                for (unsigned int k = 0; k < replacedLetters.length(); k++)
                    if (line[i] == replacedLetters.at(k))
                        line[i] = L'?';
            }
        }
    }
    outdata << line << endl;
}

outdata << endl;

if (number % 2 == 0)
    outdata << L"Матері" << endl << L"В хаті сонячній промінь косо" << endl
    << L"На долівку ляга з вікна..." << endl << L"Твої чорні шовкові коси" << endl <<
    L"Припорошила вже сивина." << endl;
else
    outdata << L"Вклонися їй" << endl << L"Коли малим ти вперше став на
ноги -" << endl << L"Яка ж то радість матері була!" << endl << L"Від тихої колиски
до порога" << endl << L"Вона тебе за руку провела." << endl;

indata.close();
outdata.close();

return 0;
}

int t10_2(string file) {
    wstring line;
    wchar_t character;
    time_t rawtime;
    time(&rawtime);
    int number = 0;

    wifstream indata;
    indata.open(file);
    wofstream outdata;
    outdata.open(file, ios_base::app);

```

```

    if (!indata || !outdata)
        return 1;

    indata.imbue(locale(locale(), new codecvt_utf8<wchar_t>));
    outdata.imbue(locale(locale(), new codecvt_utf8<wchar_t>));

    while (indata >> character) {
        number += 1;
        if (number % 2 != 0)
            line += character;
    }

    outdata << endl << L"Непарні символи речення: " << line << endl;
    outdata << L"Дата й час дозапису: " << ctime(&rawtime);

    indata.close();
    outdata.close();

    return 0;
}

int t10_3(string file, int x, int y, int z, int b) {
    time_t rawtime;
    time(&rawtime);

    ofstream data;
    data.open(file, ios_base::app);

    if (!data)
        return 1;

    data << endl << "s = " << s_calculation(x, y, z) << endl;
    if (b > 0) {
        string binary = bitset<numeric_limits<int>::digits>(b).to_string();
        binary.erase(0, binary.find_first_not_of('0'));
        data << "b у двійковому коді: " << binary << endl;
    }
    else
        data << "b — не натуральне число" << endl;

    data.close();
    return 0;
}

```

Лістинг вихідного коду із TestDriver

```

#include "ModulesYasinitskiy.h"
#include <iostream>
#include <locale>
#include <fstream>
#include <codecvt>
#include <string>
#include <limits>

using namespace std;

string input = "input.txt";
string output = "output.txt";

bool createInput(wstring content) {
    wofstream data;
    data.open(input);

    data.imbue(locale(locale(), new codecvt_utf8<wchar_t>));

    if (!data) {
        wcout << L"Помилка запису вхідних даних." << endl;
        return false;
    }

    data << content << endl;

    return true;
}

bool test1_1() {
    wstring line;
    wstring lines[3] = {L"Розробник: Yasinitskiy Yan", L"Установа/організація:
Центральноукраїнський національний технічний університет", L"Рік розробки:
2023"};
    bool linesFound[3] = {false, false, false};
    int currentLine = 0;
    if(!createInput(L"тест"))
        return false;

    t10_1(input, output);

    wifstream indata;
    indata.open(output);

    indata.imbue(locale(locale(), new codecvt_utf8<wchar_t>));

```

```

while (getline(indata,line)) {
    if (line.find(lines[currentLine]) != wstring::npos) {
        linesFound[currentLine] = true;
        currentLine++;
    }
    if (linesFound[0] && linesFound[1] && linesFound[2])
        return true;
}
return false;
}

bool test1(wstring data, wstring modified, bool even) {
    wstring line;
    bool modifiedCorrect = false;

    if(!createInput(data))
        return false;

    t10_1(input, output);

    wifstream indata;
    indata.open(output);

    indata.imbue(locale(locale(), new codecvt_utf8<wchar_t>));

    while (getline(indata,line)) {
        if (line.find(L"Речения: " + modified) != wstring::npos)
            modifiedCorrect = true;
        if (modifiedCorrect && ((even && line.find(L"Матери") != wstring::npos) ||
(!even && line.find(L"Вклонися їй") != wstring::npos)))
            return true;
    }

    return false;
}

bool test2_1() {
    wstring line;
    if(!createInput(L"тест"))
        return false;

    t10_2(input);

    wifstream indata;
    indata.open(input);

    indata.imbue(locale(locale(), new codecvt_utf8<wchar_t>));

```

```

while (getline(indata,line)) {
    if (line.find(L"Дата й час дозапису: ") != wstring::npos)
        return true;
}

return false;
}

bool test2(wstring data, wstring oddCharacters) {
    wstring line;
    if(!createInput(data))
        return false;

    t10_2(input);

    wifstream indata;
    indata.open(input);

    indata.imbue(locale(locale(), new codecvt_utf8<wchar_t>));

    while (getline(indata,line)) {
        if (line.find(L"Непарні символи речення: " + oddCharacters) !=
wstring::npos)
            return true;
    }

    return false;
}

bool test3(int x, int y, int z, int b, float s, wstring bin) {
    wstring line;
    bool firstValid = true;
    t10_3(output, x, y, z, b);

    wifstream indata;
    indata.open(output);

    indata.imbue(locale(locale(), new codecvt_utf8<wchar_t>));

    while (getline(indata,line)) {
        string s_str = to_string(s);
        wstring s_wstr (s_str.begin(), s_str.end());
        if (!firstValid && line.find(L"s = " + s_wstr) != wstring::npos)
            firstValid = true;
        else if (firstValid && line.find(L"b у двійковому коді: " + bin) !=
wstring::npos)
            return true;
    }
}

```



```

    }

    return false;
}

int main() {
    const wstring data1_1[] = {L"Тест.", L"Тест123.", L"Непарне."};
    const wstring data1_2[] = {L"Тес?.", L"Тес?123.", L"Непарне."};
    bool even[] = {true, true, false};

    const wstring data2_1[] = {L"Тест.", L"Тест123.", L"Непарне."};
    const wstring data2_2[] = {L"Тс.", L"Тс13", L"Нпре"};

    const int x[] = {10, 0, 4, 4};
    const int y[] = {2, 1, 1, 0};
    const int z[] = {5, 1, 1, 0};
    const int b[] = {1, 3, 8, 15};
    const float s[] = {9.49993801116943359375, numeric_limits<float>::quiet_NaN(),
2.0536382198333740234375, 1};
    const wstring bin[] = {L"1", L"11", L"1000", L"1111"};

    setlocale(LC_ALL, "");

    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 4; j++) {
            bool value = false;
            if (i < 2 && j == 0) {
                if (i == 0)
                    value = test1_1();
                else if (i == 1)
                    value = test2_1();
            }
            else if (i == 0)
                value = test1(data1_1[j-1], data1_2[j-1], even[j-1]);
            else if (i == 1)
                value = test2(data2_1[j-1], data2_2[j-1]);
            else if (i == 2)
                value = test3(x[j], y[j], z[j], b[j], s[j], bin[j]);
            cout << "Test " << i + 1 << "." << j + 1 << " " << (value ? "passed" :
"failed") << endl;
        }
    }

    return 0;
}

```

Назва тестового набору Test Suite Description	Задача 10.1
Назва проекту / ПЗ Name of Project / Software	Лабораторна робота 10
Рівень тестування Level of Testing	модульний / Unit Testing
Автор тест-сююта Test Suite Author	Yasinitskiy Yan
Виконавець Implementer	Yasinitskiy Yan

Ід-р тест- кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування (пройшов/не вдалося/ заблокований) / Test Result (passed/failed/ blocked)
1.1	Довільні вхідні дані	У вихідному файлі є повідомлення: Розробник: Yasinitskiy Yan Установа/організація: Центральноукраїнський національний технічний університет Місто: Кропивницький Країна: Україна Рік розробки: 2023	passed
1.2	Тест.	Тес?. «Матері»	passed
1.3	Тест123.	Тес?123. «Матері»	passed
1.4	Непарне.	Непарне. «Вклонися їй»	passed

Назва тестового набору Test Suite Description	Задача 10.2
Назва проекту / ПЗ Name of Project / Software	Лабораторна робота 10
Рівень тестування Level of Testing	модульний / Unit Testing
Автор тест-сюита Test Suite Author	Yasinitskiy Yan
Виконавець Implementer	Yasinitskiy Yan

Ід-р тест- кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування (пройшов/не вдалося/ заблокований) / Test Result (passed/failed/ blocked)
2.1	Довільні вхідні дані	У вхідному файлі є повідомлення: Дата й час дозапису: ДТЖ МММ ДД гг:хх:сс РРРР де: ДТЖ — день тижня МММ — назва місяця ДД — число (одна або дві цифри) гг:хх:сс — години, хвилини, секунди відповідно РРРР — рік Дата і час мають відповідати тим, що встановлені на комп'ютері.	passed
2.2	Тест.	Тс.	passed
2.3	Тест123.	Тс13	passed
2.4	Непарне.	Нпне	passed

Назва тестового набору Test Suite Description	Задача 10.3
Назва проекту / ПЗ Name of Project / Software	Лабораторна робота 10
Рівень тестування Level of Testing	модульний / Unit Testing
Автор тест-сьюта Test Suite Author	Yasinitskiy Yan
Виконавець Implementer	Yasinitskiy Yan

Ід-р тест-кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування (пройшов/не вдалося/ заблокований) / Test Result (passed/failed/ blocked)
3.1	x = 10, y = 2, z = 5, b = 1	s = 9.49993801116943359375 b у двійковому коді: 1	passed
3.2	x = 0, y = 1, z = 1, b = 3	s = NaN b у двійковому коді: 11	passed
3.3	x = 4, y = 1, z = 1, b = 8	s = 2.0536382198333740234375 b у двійковому коді: 1000	passed
3.4	x = 4, y = 0, z = 0, b = 15	s = 1 b у двійковому коді: 1111	passed

Висновок test driver:

Test 1.1 passed

Test 1.2 passed

Test 1.3 passed

Test 1.4 passed

Test 2.1 passed

Test 2.2 passed

Test 2.3 passed

Test 2.4 passed

Test 3.1 passed

Test 3.2 passed

Test 3.3 passed

Test 3.4 passed

Висновок

У даній лабораторній роботі №10 було проведено дослідження процесу реалізації програмних модулів оброблення даних складових типів з файловим введенням/виведенням. За результатами виконаної роботи, було встановлено, що для ефективної роботи з даними складових типів необхідно використовувати відповідні програмні модулі, які забезпечують оптимальну обробку даних.

У процесі дослідження було проаналізовано різноманітні методи оброблення даних складових типів, серед яких були виокремлені такі, як введення та виведення даних з файлів, що забезпечує зручність та швидкість роботи з великими обсягами даних. Використання програмних модулів оброблення даних складових типів з файловим введенням/виведенням є необхідним для досягнення оптимальної роботи з даними та забезпечення їх ефективного зберігання та передачі.

Застосування відповідних програмних модулів дозволяє зберігати та обробляти дані складових типів з файловим введенням/виведенням, що підвищує продуктивність та швидкість роботи з великими обсягами даних. Крім того, використання таких модулів забезпечує зручність та легкість роботи з даними, оскільки вони дозволяють ефективно здійснювати їх введення та виведення з файлів.

Отже, використання програмних модулів оброблення даних складових типів з файловим введенням/виведенням є ключовим елементом для досягнення оптимальної роботи з даними. Вони допомагають забезпечити швидкість та ефективність роботи з великими обсягами даних, та зручність їх введення та виведення з файлів.