

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Центр заочної та дистанційної освіти
Кафедра кібербезпеки та програмного забезпечення

Звіт з лабораторної роботи № 11
з дисципліни “Базові методології та технології програмування”
на тему
“Реалізація програмних засобів оброблення динамічних структур даних та
бінарних файлів”

Виконав
студент академічної групи
КН-22з
Михайленко С. Г.

Перевірив
доцент
Доренський О. П.

Варіант 1

Тема: Реалізація програмних засобів оброблення динамічних структур даних та бінарних файлів

Мета роботи полягає у набутті ґрунтовних вмінь і практичних навичок командної (колективної) реалізації програмного забезпечення, розроблення функцій оброблення динамічних структур даних, використання стандартних засобів С++ для керування динамічною пам'яттю та бінарними файловими потоками.

Завдання:

1. У складі *команди* ІТ-проекта розробити програмні модулі оброблення динамічної структури даних.
2. Реалізувати програмний засіб на основі розроблених командою ІТ-проекта модулів.

Тест-сьют:

| | |
|--|----------------------------|
| Назва тестового набору Test Suite Description | Лабораторна робота 11 |
| Назва проекту / ПЗ Name of Project / Software | Лабораторна робота 11 |
| Рівень тестування Level of Testing | системний / System Testing |
| Автор тест-сьюта Test Suite Author | Михайленко Станіслав |
| Виконавець Implementer | Михайленко Станіслав |

| Ід-р тест-кейса / Test Case | Дії (кроки) / Action (Test Steps) | Очікуваний результат / Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) / Test Result (passed/failed/ |
|--------------------------------|---|--|--|
|--------------------------------|---|--|--|

| ID | | | blocked) |
|------|---|--|----------|
| 11.1 | <ol style="list-style-type: none"> Запустити застосунок Обрати додавання нового запису Додати запис: Код розділу: I Назва розділу: Живі тварини; продукти тваринного походження Код товару: 0102 21 30 00 Назва товару: корови Вийти, натиснувши 5 | <p>Зміст файла data.tsv:</p> <p>I Живі тварини; продукти тваринного походження 0102 21 30 00 корови</p> | passed |
| 11.2 | <ol style="list-style-type: none"> Запустити застосунок Обрати збереження в інший файл Ввести назву файла: test.tsv | <p>Зміст файла test.tsv відповідає змісту файла data.tsv, у разі відсутності першого файла — другий порожній</p> | passed |
| 11.3 | <ol style="list-style-type: none"> Запустити застосунок Виконати дії 1—3 тест-кейса 11.1 Шукати запис за кодом 0102 21 30 00 | <p>Результат:</p> <p>0102 21 30 00: корови, належить до розділу I (Живі тварини; продукти тваринного походження)</p> | passed |
| 11.4 | <ol style="list-style-type: none"> Запустити застосунок Виконати дії 1—3 тест-кейса 11.1 Вилучити запис 0102 21 30 00 Шукати запис за кодом 0102 21 30 00 | <p>Результат:</p> <p>Не знайдено.</p> | passed |

Лістинг коду:

ModulesMykhailenko.cpp:

```
#include <codecvt>
#include <fstream>
```

```
#include <iostream>

#include <locale>

#include <sstream>

#include <string>

#include "ModulesMykhailenko.h"

using namespace std;

vector<Product> products;

void add(vector<Product> &vec, wstring section, wstring sectionName, wstring
code, wstring name) {

    vec.push_back({});

    Product& back = vec.back();

    back.section = section;

    back.sectionName = sectionName;

    back.code = code;

    back.name = name;

}

void remove(vector<Product> &vec, wstring record) {

    bool found = false;

    for (vector<Product>::iterator it = vec.begin(); it != vec.end();) {

        if (it->code == record) {

            found = true;

            it = vec.erase(it);

        } else

            ++it;

    }

    if (found)

        wcout << L"Видалено." << endl;
```

```

else
    wcout << L"Не найдено." << endl;
}

bool read(vector<Product> &vec, string file) {
    vector<vector<wstring>> content;
    vector<wstring> row;
    wstring line, word;

    wifstream data;
    data.open(file);

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

    if (data.is_open()) {
        while(getline(data, line)) {
            row.clear();

            wstringstream str(line);

            while(getline(str, word, L'\u0009'))
                row.push_back(word);
            content.push_back(row);
        }
    } else
        return false;

    for (unsigned int i = 0; i < content.size(); i++) {
        vec.push_back({});
        Product& back = vec.back();
        back.section = content[i][0];
        back.sectionName = content[i][1];
    }
}

```

```

        back.code = content[i][2];
        back.name = content[i][3];
    }

    return true;
}

bool save(vector<Product> &vec, string file) {
    wofstream data;
    data.open(file);

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

    if (!data)
        return false;

    for (unsigned int i = 0; i < vec.size(); i++)
        data << vec[i].section << L'\u0009' << vec[i].sectionName <<
L'\u0009' << vec[i].code << L'\u0009' << vec[i].name << endl;

    return true;
}

void search(vector<Product> &vec, wstring record) {
    bool found = false;
    for (unsigned int i = 0; i < vec.size(); i++) {
        if (vec[i].code == record) {
            found = true;

            wcout << vec[i].code << L": " << vec[i].name << L", належить
до розділу " << vec[i].section << L" (" << vec[i].sectionName << L)" <<
endl;
        }
    }
}

```

```
        if (!found)
            wcout << L"Не знайдено." << endl;
    }
```

ModulesMykhailenko.h:

```
#include <vector>
#include "struct_type_project_1.h"

using namespace std;

void add(vector<Product>&, wstring, wstring, wstring, wstring);

void remove(vector<Product>&, wstring);

bool read(vector<Product>&, string);

bool save(vector<Product>&, string);

void search(vector<Product>&, wstring);
```

struct_type_project_1.h:

```
#ifndef STRUCT_TYPE_PROJECT_1_H_INCLUDED
#define STRUCT_TYPE_PROJECT_1_H_INCLUDED

#include <string>
#include <vector>

using namespace std;

struct Product {
    wstring section;
    wstring sectionName;
    wstring code;
};
```

```
        wstring name;
};

extern vector<Product> products;

#endif // STRUCT_TYPE_PROJECT_1_H_INCLUDED
```

main.cpp:

```
#include <locale>
#include <codecvt>
#include <locale>
#include <iostream>
#include "ModulesMykhailenko.h"

using namespace std;

int main() {

    ios_base::sync_with_stdio(false);
    wcout.imbue(locale(locale(), new codecvt_utf8<wchar_t>));
    wcin.imbue(locale(locale(), new codecvt_utf8<wchar_t>));
    setlocale(LC_ALL, "");

    read(products, "data.tsv");

    while (true) {
        int choice;

        wcout << L"Довідник кодів товарів згідно з УКТЗЕД" << endl <<
            L"1. Пошук запису" << endl <<
            L"2. Зберігання довідника у заданий файл" << endl <<
            L"3. Додавання нового запису у довідник" << endl <<
```



```
L"4. Вилучення запису із довідника" << endl <<  
L"5. Завершення роботи програми і запис даних" << endl <<  
L"Введіть необхідну дію: ";
```

```
wcin >> choice;  
switch (choice) {  
    case 1:  
    {  
        wstring record;  
        wcout << L"Введіть номер запису: ";  
        wcin >> record;  
        search(products, record);  
        break;  
    }  
    case 2:  
    {  
        wstring path;  
        wcout << L"Введіть шлях до файла: ";  
        wcin >> path;  
        wstring_convert<codecvt_utf8<wchar_t>, wchar_t>  
converter;  
        string converted_path = converter.to_bytes(path);  
        save(products, converted_path);  
        break;  
    }  
    case 3:  
    {  
        wstring section;  
        wstring sectionName;  
        wstring code;  
        wstring name;  
        wcout << L"Введіть код розділу: ";
```

```

        wcin >> section;
        wcout << L"Введіть назву розділу: ";
        wcin >> sectionName;
        wcout << L"Введіть код товару: ";
        wcin >> code;
        wcout << L"Введіть назву товару: ";
        wcin >> name;
        add(products, section, sectionName, code, name);
        break;
    }
case 4:
{
    wstring record;
    wcout << L"Введіть номер запису: ";
    wcin >> record;
    remove(products, record);
    break;
}
case 5:
{
    save(products, "data.tsv");
    return 0;
}
default:
    wcout << L"Помилкові дані." << endl;
}
}
}

```

Висновок: під час виконання лабораторної роботи я навчився використовувати динамічні структури даних.