

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет

ЗВІТ
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 11
з навчальної дисципліни
“Базові методології та технології програмування”
РЕАЛІЗАЦІЯ ПРОГРАМНИХ ЗАСОБІВ ОБРОБЛЕННЯ ДИНАМІЧНИХ
СТРУКТУР ДАНИХ ТА БІНАРНИХ ФАЙЛІВ

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

ВИКОНАВ
студент академічної групи КІ-22-2
Карпова Є. І.

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

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

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

Завдання:

Базові методології та технології програмування ♦ Лабораторна робота № 11

<https://github.com/odorenskyi/Tkachenko-Oleksii-KI222>
<https://github.com/odorenskyi/Titarova-Anastasiia-KI222>
<https://github.com/odorenskyi/Karpova-Yelyzaveta-KI222>

ВАРІАНТ 10

— ЗАВДАННЯ НА РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ —

Створити е-довідник кодів товарів згідно з Українською класифікацією товарів зовнішньоекономічної діяльності (номер розділу, назва розділу, група/код/ товару, назва товару).

За вибором користувача застосунок забезпечує:

- пошук запису в е-довіднику за введеним кодом товару;
- зберігання е-довідника у заданий текстовий файл;
- додавання нового запису в е-довідник;
- вилучення заданого запису з е-довідника;
- завершення роботи програми з автоматичним записом даних у файл.

Дані е-довідника автоматично завантажуються з файлу під час запуску ПЗ.

ДОВІДНИК кодів товарів згідно з Українською класифікацією товарів зовнішньоекономічної діяльності (УКТ ЗЕД)

Розділ I. Живі тварини; продукти тваринного походження >>>

- Група 01 Живі тварини
- Група 02 М'ясо та істівні субпродукти
- Група 03 Риба і ракоподібні, молюски та інші водяні безхребетні
- Група 04 Молоко та молочні продукти; яйця птиці; натуральний мед; істівні продукти тваринного походження, в іншому місці не зазначені
- Група 05 Інші продукти тваринного походження, в іншому місці не зазначені

Розділ II. Продукти рослинного походження >>>

- Група 06. Живі дерева та інші рослини; цибулини, коріння та інші аналогічні частини рослин; зрізані квіти і декоративна зелень
- Група 07. Овочі та деякі істівні коренеплоди і бульби
- Група 08. Істівні плоди та горіхи; шкірки цитрусових або динь
- Група 09. Кава, чай, мате, або парагвайський чай, прянощі
- Група 10. Зернові культури
- Група 11. Пролукція борошномельно-круп'яної промисловості: солоні; крохмалі; інчулін; пшенична

...

Український класифікатор товарів ЗЕД

I (з 01 по 05) Живі тварини; продукти тваринного походження

Код товару	Найменування товару
01	Живі тварини
02	М'ясо та істівні субпродукти
03	Риба і ракоподібні, молюски та інші водяні безхребетні
04	Молоко та молочні продукти; яйця птиці; натуральний мед; істівні продукти тваринного походження, в іншому місці не зазначені
05	Інші продукти тваринного

Склад команди ІТ-проекта:

Карпова Єлизавета, Ткаченко Олексій, Тітарова Анастасія

Варіант 10

Лістинг коду із ModulesKarpova.cpp

```
#include <codecvt>
#include <iostream>
#include <locale>
#include <string>
#include "ModulesKarpova.h"

using namespace std;

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;
}

void output(vector<Product> &vec) {
    for (unsigned int i = 0; i < vec.size(); i++)
        wcout << vec[i].code << L": " << vec[i].name << L", належить до розділу " <<
vec[i].section << L" (" << vec[i].sectionName << L)" << endl;
}
```

Лістинг із ModulesKarpova.h

```
#ifndef MODULESKARPOVA_H_INCLUDED
#define MODULESKARPOVA_H_INCLUDED

#include <vector>
#include "struct_type_project_10.h"

using namespace std;

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

void output(vector<Product>&);

#endif // MODULESKARPOVA_H_INCLUDED
```

Лістинг коду із ModulesTitarova.cpp

```
#include <codecvt>
#include <iostream>
#include <locale>
#include <string>
#include "ModulesTitarova.h"

using namespace std;
```

```

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 знайдено." << endl;
}

```

Лістинг із ModulesTitarova.h

```

#ifndef MODULESTITAROVA_H_INCLUDED
#define MODULESTITAROVA_H_INCLUDED

#include <vector>
#include "struct_type_project_10.h"

using namespace std;

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

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

#endif // MODULESTITAROVA_H_INCLUDED

```

Лістинг коду із ModulesTkachenko.cpp

```

#include <codecvt>
#include <fstream>
#include <iostream>
#include <locale>
#include <sstream>
#include <string>
#include "ModulesTkachenko.h"

using namespace std;

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;
}

```

Лістинг із ModulesTkachenko.h

```

#ifndef MODULESTKACHENKO_H_INCLUDED
#define MODULESTKACHENKO_H_INCLUDED

#include <vector>
#include "struct_type_project_10.h"

using namespace std;

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

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

#endif // MODULESTKACHENKO_H_INCLUDED

```

Лістинг коду із main.cpp

```

#include <clocale>
#include <codecvt>
#include <locale>
#include <iostream>
#include "ModulesTkachenko.h"
#include "ModulesTitarova.h"
#include "ModulesKarpova.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, "");

    vector<Product> products;

    read(products, "data.tsv");

    while (true) {
        int choice;

        wcout << L"Довідник кодів товарів згідно з УКТЗЕД" << endl <<
            L"1. Пошук запису" << endl <<
            L"2. Виведення бази даних на екран" << endl <<
            L"3. Зберігання довідника у заданий файл" << endl <<
            L"4. Додавання нового запису у довідник" << endl <<
            L"5. Вилучення запису із довідника" << endl <<
            L"6. Завершення роботи програми і запис даних" << endl <<
            L"Введіть необхідну дію: ";

        wcin >> choice;
        wcin.ignore();
        switch (choice) {
            case 1:
            {
                wstring record;
                wcout << L"Введіть номер запису: ";
                getline(wcin, record);
                search(products, record);
                break;
            }
            case 2:
            {
                output(products);
                break;
            }
            case 3:
            {
                wstring path;
                wcout << L"Введіть шлях до файла: ";
                getline(wcin, path);
                wstring_convert<codecvt_utf8<wchar_t>, wchar_t> converter;
                string converted_path = converter.to_bytes(path);
                save(products, converted_path);
                break;
            }
            case 4:
            {
                wstring section;
                wstring sectionName;
                wstring code;
                wstring name;
                wcout << L"Введіть код розділу: ";
                getline(wcin, section);
                wcout << L"Введіть назву розділу: ";
                getline(wcin, sectionName);
                wcout << L"Введіть код товару: ";
                getline(wcin, code);
                wcout << L"Введіть назву товару: ";
                getline(wcin, name);
                add(products, section, sectionName, code, name);
                break;
            }
        }
    }
}

```

```

        case 5:
        {
            wstring record;
            wcout << L"Введіть номер запису: ";
            getline(wcin, record);
            remove(products, record);
            break;
        }
        case 6:
        {
            save(products, "data.tsv");
            return 0;
        }
        default:
            wcout << L"Помилкові дані." << endl;
    }
}
}

```

Лістинг із struct_type_project_10.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

```

TestSuite

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	Karpova Elizaveta
Виконавець Implementer	Karpova Elizaveta

Ід-п 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)
11.1	1. Запустити застосунок 2. Обрати додавання нового запису 3. Додати запис: Код розділу: I Назва розділу: Живі тварини; продукти тваринного походження Код товару: 0102 21 30 00 Назва товару: корови 4. Вийти, натиснувши 6	Зміст файла data.tsv: I Живі тварини; продукти тваринного походження 0102 21 30 00 корови	passed
11.2	1. Запустити застосунок 2. Обрати збереження в інший файл 3. Ввести назву файла: test.tsv	Зміст файла test.tsv відповідає змісту файла data.tsv, у разі відсутності першого файла — другий порожній	passed
11.3	1. Запустити застосунок 2. Виконати дії 1—3 тест-кейса 11.1 3. Шукати запис за кодом 0102 21 30 00	Результат: 0102 21 30 00: корови, належить до розділу I (Живі тварини; продукти тваринного походження)	passed
11.4	1. Запустити застосунок 2. Виконати дії 1—3 тест-кейса 11.1 3. Вилучити запис 0102 21 30 00 4. Шукати запис за кодом 0102 21 30 00	Результат: Не знайдено.	passed

Висновок:

В результаті виконання лабораторної роботи з теми "Реалізація програмних засобів оброблення динамічних структур даних та бінарних файлів" я набула ґрунтовних вмінь і практичних навичок командної реалізації програмного забезпечення. Під час роботи в складі команди Іт-проекта ми успішно розробили програмні моделі для оброблення динамічної структури даних.

Нашим завданням було розробити функції оброблення динамічних структур даних, а також використати стандартні засоби мови програмування C++ для керування динамічною пам'яттю та бінарними файловими потоками. Під час роботи ми детально вивчили принципи роботи з динамічними структурами даних, такими як списки, черги або дерева, і вміли застосовувати їх для розв'язання конкретних

завдань. Крім того, під час реалізації даної лабораторної роботи ми ознайомилися зі стандартними засобами C++, які дозволяють керувати динамічною пам'яттю та бінарними файловими потоками. Це дозволило нам ефективно виконувати операції збереження та відновлення даних, а також забезпечити ефективне використання ресурсів системи.

Завдяки цій лабораторній роботі я отримала практичний досвід роботи з динамічними структурами даних та бінарними файловими потоками, що дуже корисно для мого професійного розвитку.