

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

Завдання видав  
доцент  
Доренський О. П.

Виконали  
студенти  
Кічура М.Г. (КН-22)  
Спринчан Є. С. (КБ-22-2)  
Дяченко Р. П. (КІ-22-1)

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

**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.

Базові методології та технології програмування • Лабораторна робота № 11

<https://github.com/odorenskyi/Kichura-Maksym-KN22>  
<https://github.com/odorenskyi/Diachenko-Ruslan-KI221>  
<https://github.com/odorenskyi/Sprynchan-Yelyzaveta-KB222>

## ВАРІАНТ 11

— ЗАВДАННЯ НА РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ —

Створити базу даних “**Деканат: облік студентів**”.

За вибором користувача (диспетчера) додаток ОС Windows забезпечує:

- виведення всієї бази на екран або у текстовий файл;
- додавання нового запису в базу;
- пошук запису за введеним диспетчером прізвищем студента;
- видалення заданого оператором запису з бази;
- завершення роботи програми з автоматичним записом бази у файл.

База даних автоматично завантажується з файлу під час запуску додатка (програми).

ЗАТВЕРДЖЕНО  
Наказ Міністерства освіти і  
науки України  
02.07.2015 № 705

Форма № Н-2.03

(показати найменування вищого навчального закладу)

Інститут, факультет, відділення	
Рівень вищої освіти/освітньо-кваліфікаційний рівень	
(вказати в спеціальності, бакалавр, спеціаліст, магістр)	
Напрямок підготовки	(шифр і назва)
Спеціальність	(шифр і назва)
Спеціалізація	(назва)

**НАВЧАЛЬНА КАРТКА СТУДЕНТА**

1. Прізвище, ім'я, по батькові \_\_\_\_\_

2. Дата народження \_\_\_\_\_

3. Місце народження \_\_\_\_\_

4. Громадянство \_\_\_\_\_ 5. Закінчив(ла) \_\_\_\_\_

(написати повне навчальне закладу) в \_\_\_\_\_ році

6. Сімейний стан \_\_\_\_\_

7. Місце проживання/місце реєстрації \_\_\_\_\_

(вказати індекс, область, район, назву населеного пункту, вулиця, номер будинку, квартира, телефону)

8. Наявність підпису при вступі \_\_\_\_\_

9. Зархований(а) наказом від “ \_\_\_\_\_ ” \_\_\_\_\_ 20 \_\_\_\_\_ року № \_\_\_\_\_

Лістинг коду:

ModulesDiachenko/main.cpp:

```

#include <codecvt>

#include <fstream>

#include <iostream>

#include <locale>

#include <sstream>

#include <string>

#include "ModulesDiachenko.h"


using namespace std;


bool read(vector<Student> &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({});

            Student& back = vec.back();

            back.surname = content[i][0];

            back.name = content[i][1];

            back.patronymic = content[i][2];

            back.group = content[i][3];

        }

        return true;
    }

bool save(vector<Student> &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].surname << L'\u0009' << vec[i].name << L'\u0009' <<
        vec[i].patronymic << L'\u0009' << vec[i].group << L'\u0009' << endl;

    return true;
}

```

## ModulesDiachenko.h:

```

#ifndef MODULESDIACHENKO_H_INCLUDED
#define MODULESDIACHENKO_H_INCLUDED

```

```

#include <vector>

#include "struct_type_project_11.h"


using namespace std;


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


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


#endif // MODULESDIACHENKO_H_INCLUDED

```

### ModulesKichura/main.cpp:

```

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


using namespace std;


void add(vector<Student> &vec, wstring surname, wstring name, wstring
patronymic, wstring group) {

    vec.push_back({}); Student&
    back = vec.back();

    back.surname = surname;

    back.name = name;

    back.patronymic = patronymic;

    back.group = group;

}

```

```

void remove(vector<Student> &vec, wstring surname, wstring name, wstring
patronymic) {

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

        if (it->surname == surname && it->name == name && it->patronymic ==
patronymic) {

            it = vec.erase(it);

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

            return;

        } else

            ++it;

    }

    wcout << L"Не знайдено." << endl;

}

```

### ModulesKichura.h:

```

#ifndef MODULESKICHURA_H_INCLUDED
#define MODULESKICHURA_H_INCLUDED

#include <vector>

#include "struct_type_project_11.h"

using namespace std;

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

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

#endif // MODULESKICHURA_H_INCLUDED

```

### ModulesSprynchan/main.cpp:

```

#include <codecvt>

#include <fstream>

#include <iostream>

```

```

#include <locale>

#include <sstream>

#include <string>

#include "ModulesSprynchan.h"

using namespace std;

void search(vector<Student> &vec, wstring surname) {

    bool found = false;

    for (unsigned int i = 0; i < vec.size(); i++) {

        if (vec[i].surname == surname) {

            found = true;

            wcout << vec[i].surname << L'\u0009' << vec[i].name << L'\u0009' << vec[i].patronymic << L'\u0009' << vec[i].group << L'\u0009' << endl;

        }

    }

    if (!found)

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

}

void output(vector<Student> &vec) {

    for (unsigned int i = 0; i < vec.size(); i++)

        wcout << vec[i].surname << L'\u0009' << vec[i].name << L'\u0009' << vec[i].patronymic << L'\u0009' << vec[i].group << L'\u0009' << endl;

}

```

### ModulesSprynchan.h:

```

#ifndef MODULESSPRYNCHAN_H_INCLUDED

#define MODULESSPRYNCHAN_H_INCLUDED

#include <vector>

#include "struct_type_project_11.h"

```

```
using namespace std;

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

void output(vector<Student>&);

#endif // MODULELESSPRYNCHAN_H_INCLUDED
```

### prj\_11\_Sprynchan/main.cpp:

```
#include <codecvt>
#include <locale>
#include <iostream>
#include "ModulesKichura.h"
#include "ModulesDiachenko.h"
#include "ModulesSprynchan.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>));

    vector<Student> students;

    read(students, "data.tsv");

    while (true) {
        int choice;

        wcout << L"Деканат: облік студентів" << endl <<
            L"1. Пошук запису" << endl <<
```



```
L"2. Виведення бази даних на екран" << endl <<
L"3. Зберігання бази даних у заданий файл" << endl <<
L"4. Додавання нового запису в базу" << endl <<
L"5. Вилучення запису із бази" << 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 прізвище: ";

        getline(wcin, record);

        search(students, record);

        break;

    }

    case 2:

    {

        output(students);

        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(students, converted_path);

        break;

    }

}
```

```

    }

case 4:
{
    wstring surname;

    wstring name;

    wstring patronymic;

    wstring group;


    wcout << L"Введіть прізвище: ";

    getline(wcin, surname);

    wcout << L"Введіть ім'я: ";

    getline(wcin, name);

    wcout << L"Введіть по батькові: ";

    getline(wcin, patronymic);

    wcout << L"Введіть групу: ";

    getline(wcin, group);

    add(students, surname, name, patronymic, group);

    break;
}

case 5:
{

    wstring surname;

    wstring name;

    wstring patronymic;


    wcout << L"Введіть прізвище: ";

    getline(wcin, surname);

    wcout << L"Введіть ім'я: ";

    getline(wcin, name);

    wcout << L"Введіть по батькові: ";

    getline(wcin, patronymic);

    remove(students, surname, name, patronymic);

```

```

        break;
    }
    case 6:
    {
        save(students, "data.tsv");
        return 0;
    }
    default:
        wcout << L"Помилкові дані." << endl;
    }
}
}

```

### struct\_type\_project\_11.h:

```

#ifndef STRUCT_TYPE_PROJECT_11_H_INCLUDED
#define STRUCT_TYPE_PROJECT_11_H_INCLUDED

#include <string>
#include <vector>

using namespace std;

struct Student {
    wstring surname;
    wstring name;
    wstring patronymic;
    wstring group;
};

extern vector<Student> students;

#endif // STRUCT_TYPE_PROJECT_11_H_INCLUDED

```

**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	Кічура Максим
Виконавець Implementer	Кічура Максим

Ід-р тест- кейс а / 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, натиснувши 4 3. Додати запис: Прізвище:  Кічура Ім'я:  Максим  По батькові: Григорович  Група: КН-22  4. Вийти, натиснувши 6	Зміст файла data.tsv:  Кічура Максим Григорович КН-22	passed

11.2	<ol style="list-style-type: none"> <li>1. Запустити застосунок</li> <li>2. Обрати збереження в інший файл, натиснувши 3</li> <li>3. Ввести назву файла: test.tsv</li> </ol>	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 прізвищем Кічура	Результат: Кічура Максим Григорович КН-22	passed
11.4	1. Запустити застосунок 2. Виконати дії 1 — 3 тест-кейса 11.1 3. Вилучити запис: Прізвище: Кічура  Ім'я: Максим  По батькові: Григорович  4. Шукати запис за прізвищем Кічура	Результат: Не знайдено.	passed

**Висновок:** під час виконання лабораторної роботи ми навчилися працювати над створенням програмного продукту в команді, розподіляючи написання окремих частин коду між учасниками. Також ми навчилися використовувати динамічні структури даних.