

Міністерство освіти і науки України  
Національний університет “Львівська політехніка”  
Інститут телекомунікацій, радіоелектроніки та електронної техніки



Звіт до лабораторної роботи №7  
З дисципліни : «Об’єктно орієнтоване програмування»  
На тему: « Робота з динамічними масивами »  
Варіант: 3

Виконав:  
Ст.гр АП-21  
Братейко Вадим

Прийняв:  
Алтунін С.І

**Мета роботи:** Створити, відлагодити та протестувати програму, у якій створити клас для обробки даних. Одним з членів класу є масив структур, в якому зберігаються введені дані.

### Хід роботи:

### Варіант завдань:

3	Іспит	Ім'я студента, номер білета, оцінка
---	-------	-------------------------------------

### Код програми.

Main.cpp

```
#include "student.h"
```

```
int main() {
    Student group;

    // Додавання студентів
    group.addStudent(1, "Нестор Руслан", 1);
    group.addStudent(2, "Брезвин Ростислав", 3);
    group.addStudent(3, "Савчак Роман", 2);

    // Виведення списку студентів
    std::cout << "\nПоточний список студентів:" << std::endl;
    group.printStudents();

    // Оновлення курсу студента
    std::cout << "\nОновлення курсу студента з ID 2:" << std::endl;
    group.updateStudentCourse(2, 4);

    // Видалення студента
    std::cout << "\nВидалення студента з ID 1:" << std::endl;
    group.removeStudent(1);

    // Виведення оновленого списку студентів
    std::cout << "\nОновлений список студентів:" << std::endl;
    group.printStudents();

    // Очищення списку студентів
    std::cout << "\nОчищення списку студентів:" << std::endl;
    group.clearStudents();

    return 0;
}
```

Student.cpp

```
#include "student.h"
```

```
// Конструктор
Student::Student() {}
```

```
// Деструктор
```

```

Student::~~Student() {
    clearStudents();
}

// Додавання студента
void Student::addStudent(int id, const std::string& name, int course) {
    if (course <= 0 || course > 6) {
        std::cout << "Помилка: Невалідний номер курсу!" << std::endl;
        return;
    }
    students.emplace_back(id, name, course);
}

// Видалення студента за ID
bool Student::removeStudent(int id) {
    for (auto it = students.begin(); it != students.end(); ++it) {
        if (it->id == id) {
            students.erase(it);
            return true;
        }
    }

    std::cout << "Студента з ID " << id << " не знайдено." << std::endl;
    return false;
}

// Оновлення курсу студента
bool Student::updateStudentCourse(int id, int new_course) {
    if (new_course <= 0 || new_course > 6) {
        std::cout << "Помилка: Невалідний номер курсу!" << std::endl;
        return false;
    }

    for (auto& student : students) {
        if (student.id == id) {
            student.course = new_course;
            return true;
        }
    }

    std::cout << "Студента з ID " << id << " не знайдено." << std::endl;
    return false;
}

// Виведення списку студентів
void Student::printStudents() const {
    if (students.empty()) {
        std::cout << "Список студентів порожній." << std::endl;
        return;
    }
}

```

```

    }

    std::cout << "Список студентів:\n";
    for (const auto& student : students) {
        std::cout << "ID: " << student.id
            << ", Ім'я: " << student.name
            << ", Курс: " << student.course << std::endl;
    }
}

// Очищення списку студентів
void Student::clearStudents() {
    students.clear();
    std::cout << "Список студентів очищено." << std::endl;
}

```

Student.h

```

#ifndef STUDENT_H
#define STUDENT_H

```

```

#include <string>
#include <vector>
#include <iostream>

```

// Структура студента

```

struct student_s_t {
    int id;           // ID студента
    std::string name; // Ім'я студента
    int course;       // Номер курсу

    student_s_t(int student_id, const std::string& student_name, int student_course)
        : id(student_id), name(student_name), course(student_course) { }
};

```

// Клас Student

```

class Student {
private:

```

```
std::vector<student_s_t> students; // Список студентів
```

```
public:
```

```
    // Конструктор
```

```
    Student();
```

```
    // Деструктор
```

```
    ~Student();
```

```
    // Додавання студента
```

```
    void addStudent(int id, const std::string& name, int course);
```

```
    // Видалення студента за ID
```

```
    bool removeStudent(int id);
```

```
    // Оновлення курсу студента
```

```
    bool updateStudentCourse(int id, int new_course);
```

```
    // Виведення списку студентів
```

```
    void printStudents() const;
```

```
    // Очищення списку студентів
```

```
    void clearStudents();
```

```
};
```

```
#endif // STUDENT_H
```

```
Поточний список студентів:
Список студентів:
ID: 1, Ім'я: Нестор Руслан, Курс: 1
ID: 2, Ім'я: Брезвин Ростислав, Курс: 3
ID: 3, Ім'я: Савчак Роман, Курс: 2

Оновлення курсу студента з ID 2:

Видалення студента з ID 1:

Оновлений список студентів:
Список студентів:
ID: 2, Ім'я: Брезвин Ростислав, Курс: 4
ID: 3, Ім'я: Савчак Роман, Курс: 2
```

[illegible]

**Висновок:** У ході виконання лабораторної роботи було створено клас для обробки даних, який містить динамічний масив структур. Було реалізовано механізми додавання, видалення та обробки даних у масиві, а також проведено тестування функціональності класу. Робота дозволила закріпити знання з роботи з динамічною пам'яттю, створення структурованих типів даних та об'єктно-орієнтованого програмування.