

Практична робота

Виконав: Устич Максим

Група: alk43

Київ 2025р.

Структура проекта

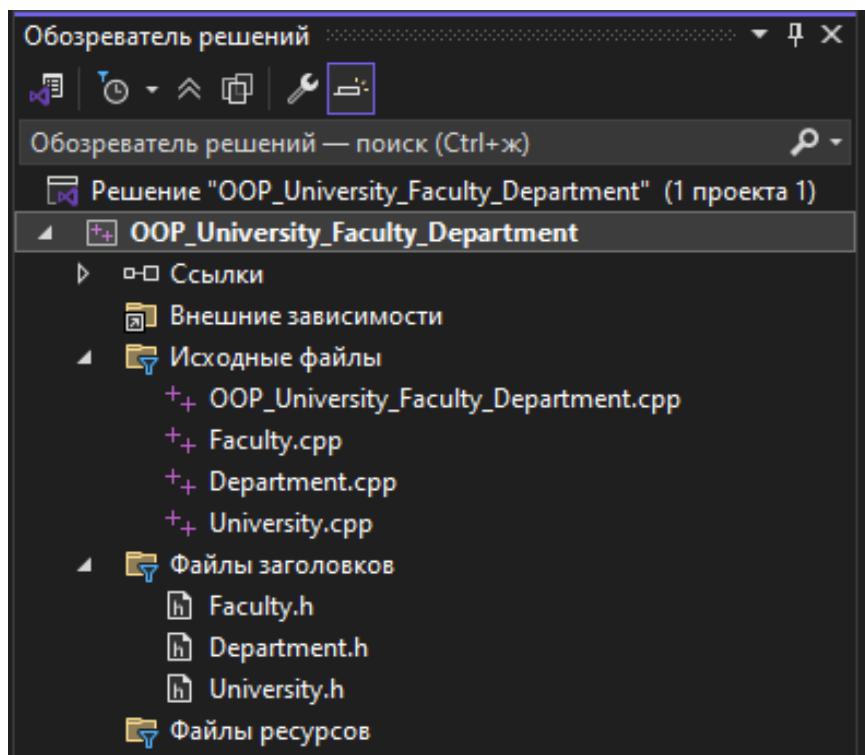


Рис. 1 – Структура проекту у Visual Studio

Код програми:

University.h

```
Department.cpp | Faculty.cpp | University.cpp | Department.h | Faculty.h | University.h* 00
+* OOP_University_Faculty_Department
  + University
    #pragma once
    #include <iostream>
    #include <string>

    class University {
    protected:
        std::string name;
        std::string city;
        int foundedYear;

    public:
        University(std::string n, std::string c, int y);

        virtual void show() const; // для поліморфізму
        virtual ~University(); // важливо: virtual деструктор
    };
```

Рис. 2 – Код файлу University.h

University.cpp

```
#include "University.h"
using namespace std;

University::University(string n, string c, int y)
    : name(n), city(c), foundedYear(y) {}

void University::show() const {
    cout << "Університет" << endl;
    cout << "Назва: " << name << endl;
    cout << "Місто: " << city << endl;
    cout << "Рік заснування: " << foundedYear << endl << endl;
}

University::~University() {}
```

Puc. 3 – Код файлу University.cpp

Faculty.h

```
#pragma once
#include "University.h"

class Faculty : public University {
protected:
    std::string facultyName;
    std::string dean;
    int students;

public:
    Faculty(std::string n, std::string c, int y,
            std::string fac, std::string d, int st);

    void show() const override; // override – важливо
    ~Faculty() override;
};
```

Puc. 4 – Код файлу Faculty.h

Faculty.cpp

```
Department.cpp Faculty.cpp* ✘ University.cpp* Department.h Faculty.h* University.h* 00
[+] OOP_University_Faculty_Department Faculty
#include "Faculty.h"
using namespace std;

Faculty::Faculty(string n, string c, int y,
                 string fac, string d, int st)
    : University(n, c, y),
      facultyName(fac), dean(d), students(st) {
}

void Faculty::show() const {
    cout << "Факультет" << endl;
    cout << "Університет: " << name << endl;
    cout << "Місто: " << city << endl;
    cout << "Рік заснування: " << foundedYear << endl;
    cout << "Назва факультету: " << facultyName << endl;
    cout << "Декан: " << dean << endl;
    cout << "К-сть студентів: " << students << endl << endl;
}

Faculty::~Faculty() {}
```

Puc. 5 – Код файла Faculty.cpp

Department.h

```
Department.cpp Faculty.cpp* University.cpp* Department.h* ✘ Faculty.h* University.h* 00
[+] OOP_University_Faculty_Department Department
#pragma once
#include "Faculty.h"

class Department : public Faculty {
    std::string departmentName;
    std::string head;
    int teachers;

public:
    Department(std::string n, std::string c, int y,
               std::string fac, std::string d, int st,
               std::string dep, std::string h, int t);

    void show() const override;
    ~Department() override;
};
```

Puc. 6 – Код файла Department.h

Department.cpp

```
Department.cpp Faculty.cpp* University.cpp* Department.h* Faculty.h* University.h* 00
+ OOP_University_Faculty_Department
    #include "Department.h"
    using namespace std;

    Department::Department(string n, string c, int y,
                           string fac, string d, int st,
                           string dep, string h, int t)
        : Faculty(n, c, y, fac, d, st),
          departmentName(dep), head(h), teachers(t) {
    }

    void Department::show() const {
        cout << "Кафедра" << endl;
        cout << "Університет: " << name << endl;
        cout << "Місто: " << city << endl;
        cout << "Рік заснування: " << foundedYear << endl;
        cout << "Факультет: " << facultyName << endl;
        cout << "Декан: " << dean << endl;
        cout << "К-сть студентів: " << students << endl;
        cout << "Кафедра: " << departmentName << endl;
        cout << "Зав. кафедри: " << head << endl;
        cout << "К-сть викладачів: " << teachers << endl;
    }

    Department::~Department() {}

```

Рис. 7 – Код файлу Department.cpp

main.cpp

```
main.cpp
    #include <iostream>
    #include <climits>
    #include <exception>

    #include "University.h"
    #include "Faculty.h"
    #include "Department.h"

    using namespace std;

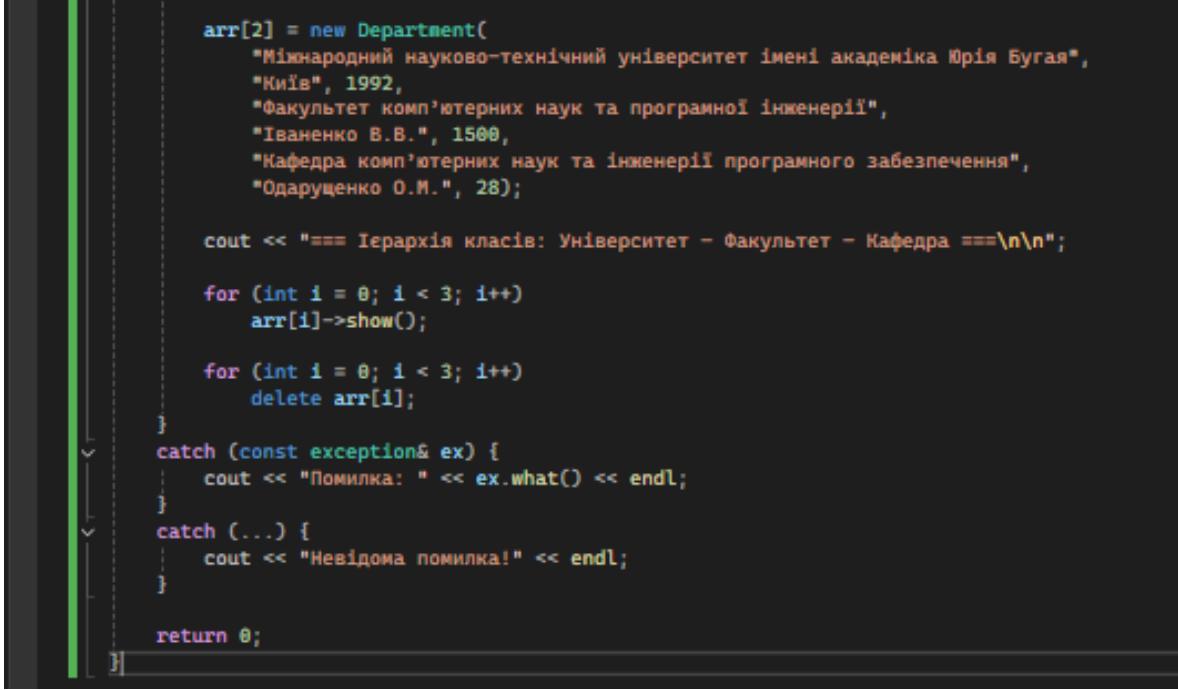
    int main() {
        setlocale(LC_ALL, "");
        system("chcp 65001 > nul");

        try {
            University* arr[3];

            arr[0] = new University(
                "Міжнародний науково-технічний університет імені академіка Юрія Бугая",
                "Київ", 1992);

            arr[1] = new Faculty(
                "Міжнародний науково-технічний університет імені академіка Юрія Бугая",
                "Київ", 1992,
                "Факультет комп'ютерних наук та програмної інженерії",
                "Іваненко В.В.", 1500); // декан – умовні дані
        }
    }

```



```
arr[2] = new Department(
    "Міжнародний науково-технічний університет імені академіка Юрія Бугая",
    "Київ", 1992,
    "Факультет комп'ютерних наук та програмної інженерії",
    "Іваненко В.В.", 1500,
    "Кафедра комп'ютерних наук та інженерії програмного забезпечення",
    "Одарущенко О.М.", 28);

cout << "==== Ієархія класів: Університет – Факультет – Кафедра ==="\n\n;

for (int i = 0; i < 3; i++)
    arr[i]->show();

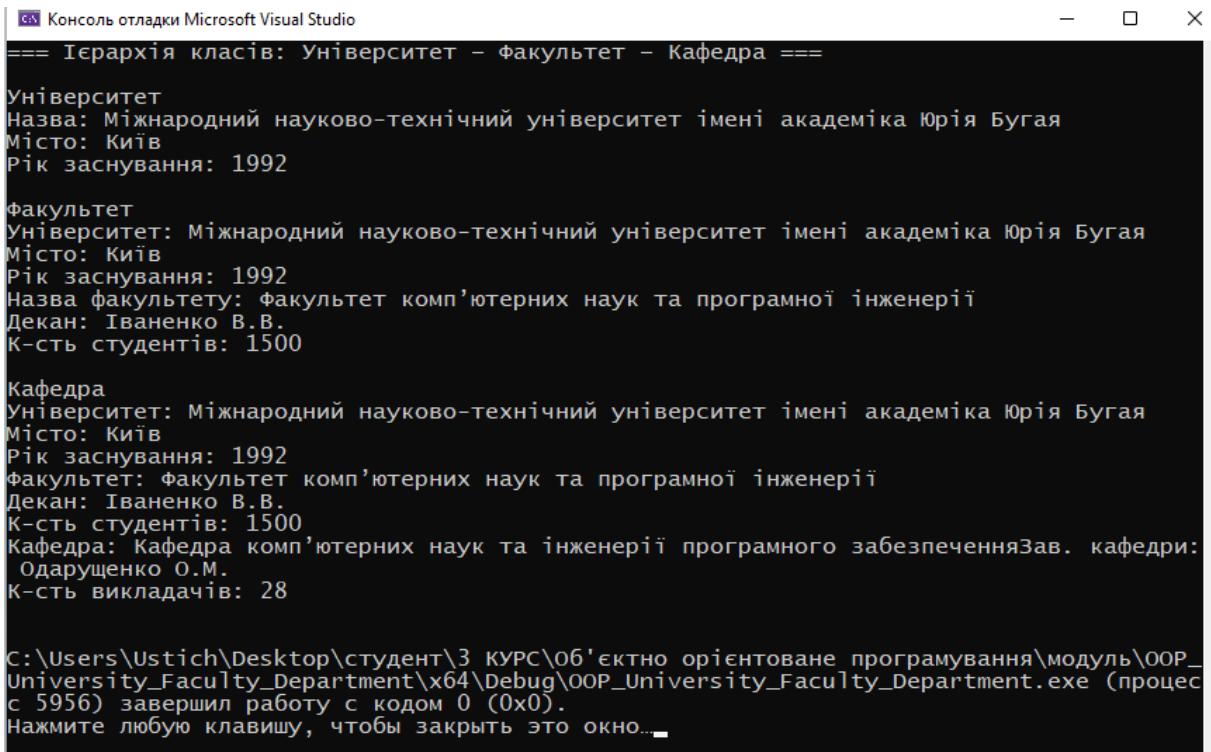
for (int i = 0; i < 3; i++)
    delete arr[i];
}

catch (const exception& ex) {
    cout << "Помилка: " << ex.what() << endl;
}
catch (...) {
    cout << "Невідома помилка!" << endl;
}

return 0;
}
```

Рис. 8 – Код файлу `main.cpp`

Результат роботи програми



```
Консоль отладки Microsoft Visual Studio
==== Ієархія класів: Університет – факультет – Кафедра ===

Університет
Назва: Міжнародний науково-технічний університет імені академіка Юрія Бугая
Місто: Київ
Рік заснування: 1992

Факультет
Університет: Міжнародний науково-технічний університет імені академіка Юрія Бугая
Місто: Київ
Рік заснування: 1992
Назва факультету: факультет комп'ютерних наук та програмної інженерії
Декан: Іваненко В.В.
К-сть студентів: 1500

Кафедра
Університет: Міжнародний науково-технічний університет імені академіка Юрія Бугая
Місто: Київ
Рік заснування: 1992
Факультет: факультет комп'ютерних наук та програмної інженерії
Декан: Іваненко В.В.
К-сть студентів: 1500
Кафедра: Кафедра комп'ютерних наук та інженерії програмного забезпечення
Зав. кафедри: Одарущенко О.М.
К-сть викладачів: 28

C:\Users\Ustich\Desktop\студент\3 КУРС\06'єктно орієнтоване програмування\модуль\OOP_University_Faculty_Department\x64\Debug\OOP_University_Faculty_Department.exe (процес с 5956) завершив роботу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно...
```

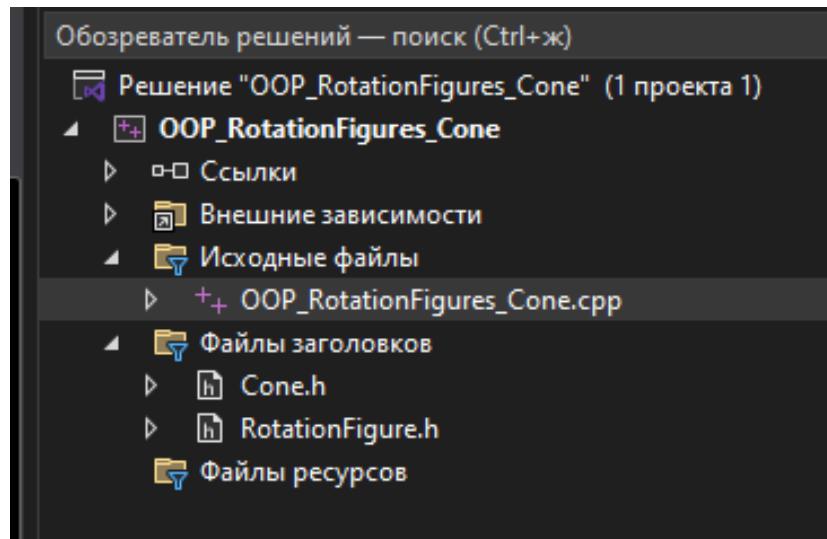
Рис. 7 – Результат роботи програми

Примітка.

Дані про декана факультету наведені умовно та використані виключно з навчальною метою.

Ієрархія класів "Фігури обертання" – "Конус".

Структура проекта.



Ruc. 8 – Структура проекту у Visual Studio

Код програми:

Rotation Figure.h

```
#pragma once
#include <iostream>
#include <string>

class RotationFigure {
protected:
    std::string name;

public:
    RotationFigure(const std::string& n) : name(n) {}

    virtual void show() const {
        std::cout << "Фігура обертання: " << name << std::endl;
    }

    virtual ~RotationFigure() {}
};
```

Рис. 9 – Код файлу RotationFigure.h

Cone.h

```
#pragma once
#include "RotationFigure.h"
#include <cmath>

// щоб M_PI точно працювало у Visual Studio
#ifndef M_PI
#define M_PI 3.14159265358979323846
#endif

class Cone : public RotationFigure {
    double r; // радіус основи
    double h; // висота

public:
    Cone(double radius, double height)
        : RotationFigure("Конус"), r(radius), h(height) {}

    double volume() const {
        return (M_PI * r * r * h) / 3.0;
    }

    double area() const {
        double l = std::sqrt(r * r + h * h); // твірна
        return M_PI * r * (r + l); // повна площа
    }

    void show() const override {
        std::cout << "Фігура обертання: " << name << std::endl;
        std::cout << "Радіус r: " << r << std::endl;
        std::cout << "Висота h: " << h << std::endl;
        std::cout << "Об'єм V: " << volume() << std::endl;
        std::cout << "Площа поверхні S: " << area() << std::endl;
    }

    ~Cone() override {}
};
```

Рис. 10 – Код файлу Cone.h

main.cpp

```
Cone.h RotationFigure.h OOP_Rotation...res_Cone.cpp (Глобальная область)
#include <iostream>
#include <clocale>
#include <exception>

#include "RotationFigure.h"
#include "Cone.h"

using namespace std;

int main() {
    setlocale(LC_ALL, "");
    system("chcp 65001 > nul");

    try {
        RotationFigure* arr[3];

        // різні типи об'єктів у масиві вказівників на базовий клас
        arr[0] = new RotationFigure("Циліндр (умовно)"); // просто базовий об'єкт
        arr[1] = new Cone(3.0, 5.0);
        arr[2] = new Cone(2.5, 4.0);

        cout << "==== Ієархія класів: фігури обертання - Конус ===\n\n";
        for (int i = 0; i < 3; i++)
            arr[i]->show();

        for (int i = 0; i < 3; i++)
            delete arr[i];
    }
    catch (const exception& ex) {
        cout << "Помилка: " << ex.what() << endl;
    }
    catch (...) {
        cout << "Невідома помилка!" << endl;
    }

    return 0;
}
```

Рис. 11 – Код файлу main.cpp

Результат роботи програми

```
Консоль отладки Microsoft Visual Studio
==== Ієархія класів: фігури обертання - Конус ===
фігура обертання: Циліндр (умовно)
фігура обертання: Конус
Радіус r: 3
Висота h: 5
Об'єм V: 47.1239
Площа поверхні S: 83.2298

фігура обертання: Конус
Радіус r: 2.5
Висота h: 4
Об'єм V: 26.1799
Площа поверхні S: 56.6821

C:\Users\Ustich\Desktop\студент\3 КУРС\06'єктно орієнтоване
програмування\модуль\OOP_RotationFigures_Cone\x64\Debug\OOP_
RotationFigures_Cone.exe (процес 5084) завершил работу с ко
дом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно...
```

Рис. 12 – Результат роботи програми

Шаблонна функцію для пошуку в масиві елементів, які знаходяться в певному діапазоні.

Структура проекту

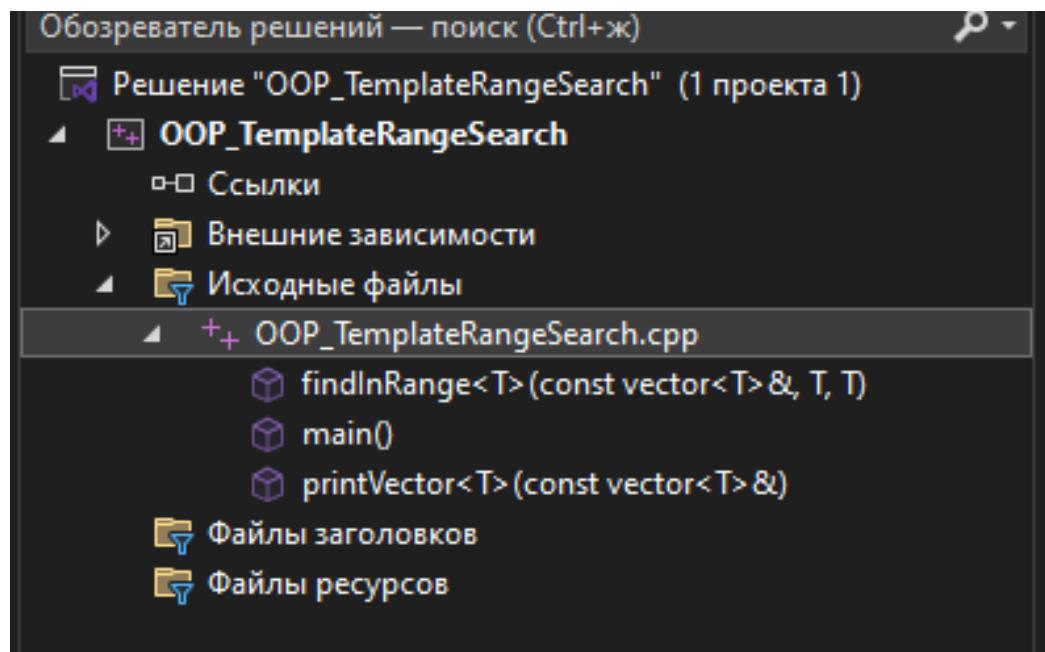


Рис. 13 – Структура проекту у Visual Studio

Код програми

```
OOP_Template...geSearch.cpp  X
+
OOP_TemplateRangeSearch
(Глобальная область)

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

using namespace std;

// =====
// Шаблонна функція пошуку елементів
// у діапазоні [low; high]
// =====
```

```

// template <typename T>
vector<T> findInRange(const vector<T>& arr, T low, T high) {
    vector<T> result;

    for (const T& el : arr) {
        if (el >= low && el <= high) {
            result.push_back(el);
        }
    }
    return result;
}

// =====
// Шаблонна функція виведення вектора
// =====
template <typename T>
void printVector(const vector<T>& v) {
    if (v.empty()) {
        cout << "Немає елементів у заданому діапазоні." << endl;
        return;
    }

    for (const T& el : v) {
        cout << el << " ";
    }
    cout << endl;
}

int main() {
    // Коректне відображення української мови
    setlocale(LC_ALL, "");
    system("chcp 65001 > nul");

    // ===== Перевірка для int =====
    vector<int> arrInt = { 1, 5, 10, 20, 50, 100 };
    cout << "int, діапазон [5; 30]:" << endl;
    printVector(findInRange(arrInt, 5, 30));
    cout << endl;

    // ===== Перевірка для double =====
    vector<double> arrDouble = { 1.1, 5.5, 7.2, 12.8, 30.3 };
    cout << "double, діапазон [5.0; 15.0]:" << endl;
    printVector(findInRange(arrDouble, 5.0, 15.0));
    cout << endl;

    // ===== Перевірка для char =====
    vector<char> arrChar = { 'a', 'b', 'c', 'd', 'x', 'z' };
    cout << "char, діапазон ['b'; 'x']:" << endl;
    printVector(findInRange(arrChar, 'b', 'x'));
    cout << endl;

    // ===== Перевірка для string =====
    vector<string> arrString = { "apple", "banana", "car", "desk", "zoo" };
    cout << "string, діапазон [\\"banana\\"; \\"desk\\"]:" << endl;
    printVector(findInRange(arrString, string("banana"), string("desk")));
    cout << endl;

    return 0;
}

```

Puc. 14 – Код файлу main.cpp

Результат роботи програми

```
Консоль отладки Microsoft Visual Studio
int, діапазон [5; 30]:
5 10 20

double, діапазон [5.0; 15.0]:
5.5 7.2 12.8

char, діапазон ['b'; 'x']:
b c d x

string, діапазон ["banana"; "desk"]:
banana car desk

C:\Users\Ustich\Desktop\студент\3 КУРС\об'єктно орієнтоване програмування\модуль\OOP_TemplateRangeSearch\x64\Debug\OOP_TemplateRangeSearch.exe (процесс 19732) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно...
```

Рис. 15 – Результат виконання програми.

Шаблон класу для зберігання пари чисел різних типів

Структура проекту

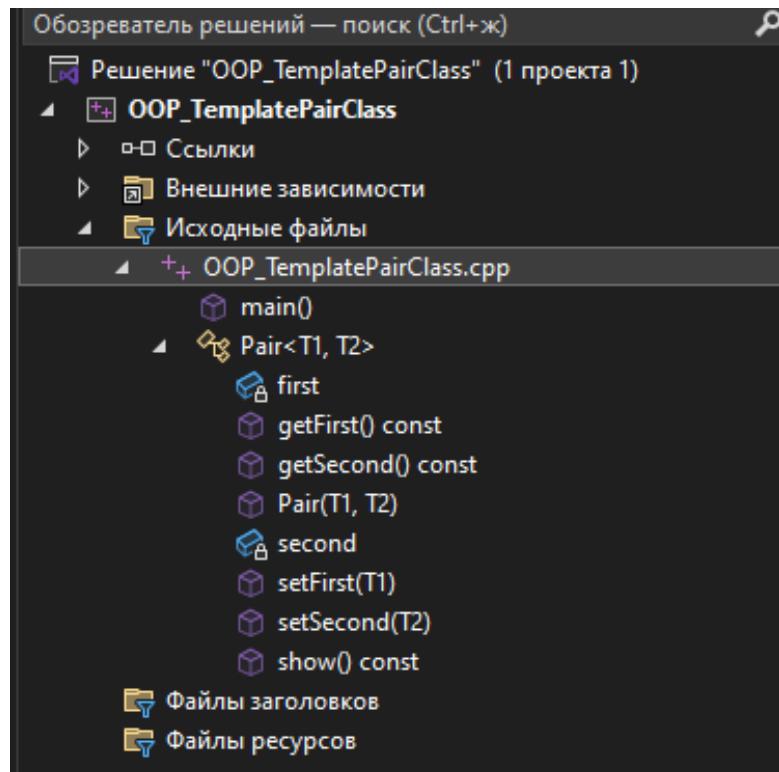


Рис. 16 – Структура проекту у Visual Studio.

Код програми

```
эль элементов  +++ OOP_TemplatePairClass  (Глобальная область)
    #include <iostream>
    #include <locale>

    using namespace std;

    // Шаблон класу Pair (пара значень різних типів)
    template <typename T1, typename T2>
    class Pair {
    private:
        T1 first;
        T2 second;

    public:
        Pair(T1 a, T2 b) : first(a), second(b) {}

        T1 getFirst() const { return first; }
        T2 getSecond() const { return second; }

        void setFirst(T1 a) { first = a; }
        void setSecond(T2 b) { second = b; }

        void show() const {
            cout << "Перше значення: " << first << endl;
            cout << "Друге значення: " << second << endl;
        }
    };

    int main() {
        // Українська вивід (як ти просив – щоб не було кракозябр)
        setlocale(LC_ALL, "");
        system("chcp 65001 > nul");

        cout << "==== Шаблон класу Pair (пара чисел) ===\n\n";

        // 1) int + double
        Pair<int, double> p1(10, 3.14);
        cout << "Пара 1 (int + double):\n";
        p1.show();
        cout << endl;

        // 2) double + int
        Pair<double, int> p2(5.75, 20);
        cout << "Пара 2 (double + int):\n";
        p2.show();
        cout << endl;

        // 3) long + float
        Pair<long, float> p3(100000, 2.5f);
        cout << "Пара 3 (long + float):\n";
        p3.show();
        cout << endl;

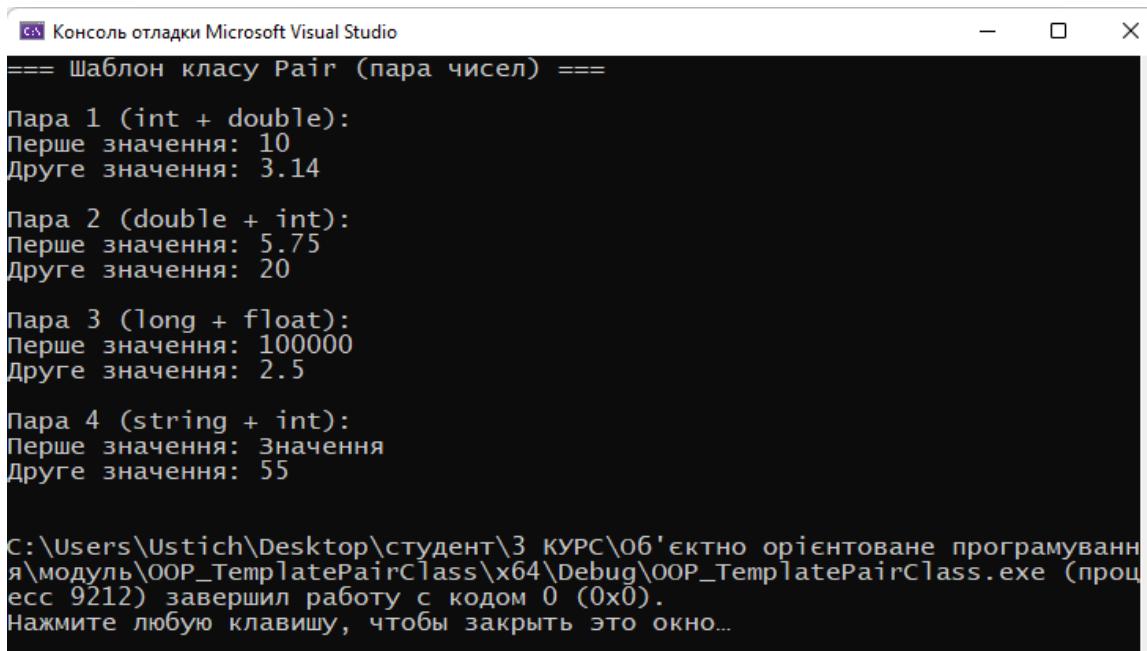
        // 4) string + int (додатково, але норм для демонстрації шаблону)
        Pair<string, int> p4("Значення", 55);
        cout << "Пара 4 (string + int):\n";
        p4.show();
        cout << endl;

        return 0;
    }

```

Рис. 17 – Код файла OOP_TemplatePairClass.cpp

Результат роботи програми



Консоль отладки Microsoft Visual Studio

```
== Шаблон класу Pair (пара чисел) ==

Пара 1 (int + double):
Перше значення: 10
Друге значення: 3.14

Пара 2 (double + int):
Перше значення: 5.75
Друге значення: 20

Пара 3 (long + float):
Перше значення: 100000
Друге значення: 2.5

Пара 4 (string + int):
Перше значення: Значення
Друге значення: 55

C:\Users\Ustich\Desktop\студент\3 КУРС'06' ектно орієнтоване програмування\модуль\OOP_TemplatePairClass\x64\Debug\OOP_TemplatePairClass.exe (процес 9212) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно...
```

Рис. 18 – Результат виконання програми