Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

**ЛАБОРАТОРНА РОБОТА №1**

**Тема:** Основні поняття ООП. Класи та об’єкти. Функції доступу. Вказівник this

# з дисципліни: Об’єктно-орієнтоване програмування

|  |
| --- |
| Виконав: ст. гр. КН-23 |
| Царенко С. О. |
| Перевірив: асистент  Козірова Н. Л. |
|  |

Кропивницький

2024

**Тема:** Основні поняття ООП. Класи та об’єкти. Функції доступу. Вказівник this

**Мета:** ознайомитись з основними поняттями ООП. Вивчити поняття клас, об’єкт, сеттер, геттер та навчитись їх програмно реалізовувати мовою С++.

**Варіант 5**

**Завдання:**

1. Створіть клас "Student" (студент), який має наступні властивості та функціональність:
2. Приватні поля класу:

• name (ім'я студента)

• age (вік студента)

• major (спеціальність студента)

1. Публічні методи класу:

• Метод setName(), який дозволяє задати ім'я студента.

• Метод getName(), який повертає ім'я студента.

• Метод setAge(), який дозволяє задати вік студента.

• Метод getAge(), який повертає вік студента.

• Метод setMajor(), який дозволяє задати спеціальність студента.

• Метод getMajor(), який повертає спеціальність студента.

1. Створіть об'єкт класу "Student".
2. Задайте значення полів об'єкта за допомогою відповідних методів.
3. Виведіть інформацію про студента на екран, використовуючи методи для отримання значень полів
4. Реалізувати програму за допомогою роздільної компіляції. У вашому рішенні можуть бути додаткові методи та поля, якщо ви вважаєте їх необхідними.
5. Реалізувати вище наведену задачу за допомогою структурного програмування. У висновку описати різницю цих методів.
6. **Student.cpp:**

#include<iostream>

#include<string>

#include "Student.h"

using namespace std;

void Student::SetName(string newName)

{

this->name = newName;

}

void Student::SetAge(int newAge)

{

this->age = newAge;

}

void Student::SetMajor(string newMajor)

{

this->major = newMajor;

}

string Student::GetName()

{

return name;

}

int Student::GetAge()

{

return age;

}

string Student::GetMajor()

{

return major;

}

1. **Student.h:**

#include<string>

#include<iostream>

using namespace std;

#pragma once

class Student

{

private:

string name;

int age;

string major;

public:

void SetName(string newName);

void SetAge(int newAge);

void SetMajor(string newMajor);

string GetName();

int GetAge();

string GetMajor();

};

1. **Source.cpp:**

#include<iostream>

#include<string>

#include "Student.h"

using namespace std;

void main() {

Student student;

string name, major;

int age;

cout << "Please enter name: " << endl;

cin >> name;

cout << "Please enter age: " << endl;

do {

cin >> age;

} while (age <= 6);

cout << "Please enter major: " << endl;

cin >> major;

student.SetName(name);

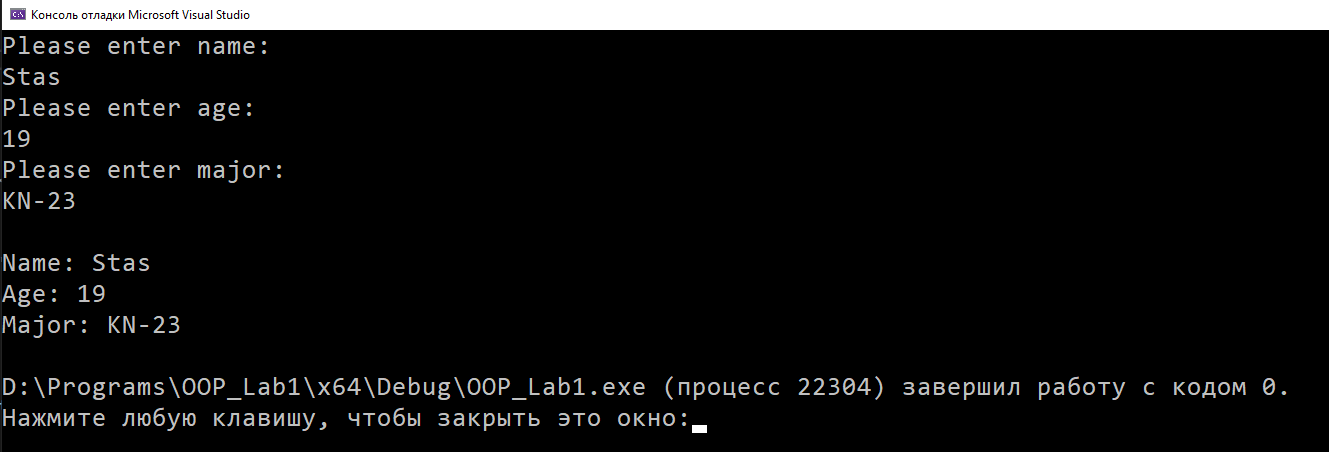
student.SetAge(age);

student.SetMajor(major);

cout << endl << "Name: " << student.GetName() << "\nAge: " << student.GetAge() << "\nMajor: " << student.GetMajor() << endl;

}

**Скрін виконання програми:**

****

**Завдання 2:**

**Task2.cpp:**

#include <iostream>

#include <string>

using namespace std;

int main() {

struct Student {

string name;

int age;

string major;

};

Student student;

student.name = "Roberto";

student.age = 25;

student.major = "KB-21";

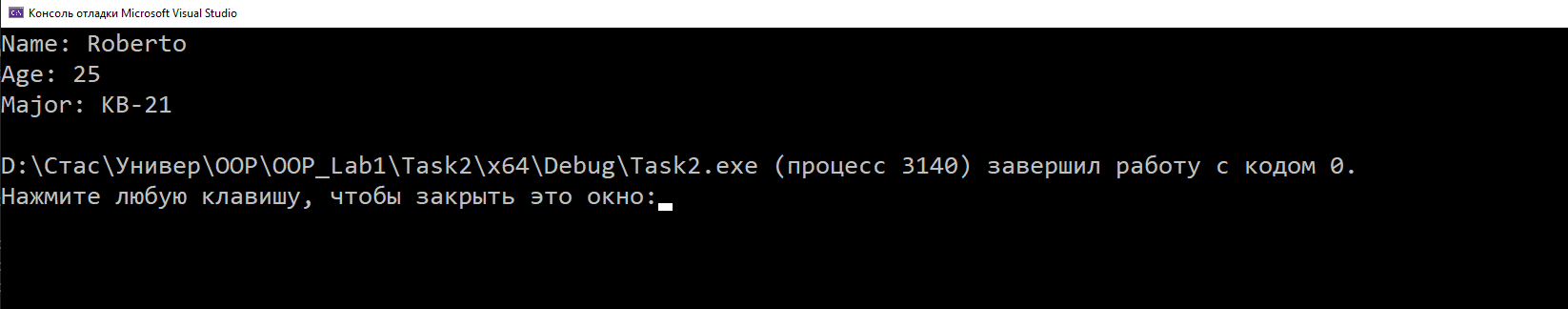
cout << "Name: " << student.name << endl;

cout << "Age: " << student.age << endl;

cout << "Major: " << student.major << endl;

}

**Скрін виконання програми:**



**Висновок:**

У ході виконання лабораторної роботи №1 “ Основні поняття ООП. Класи та об’єкти. Функції доступу. Вказівник this. ”. Я ознайомився та закріпив свої знання з основними поняттями Об’єктно-орієнтованим програмуванням (ООП). Вивчити поняття клас, об’єкт, сеттер, геттер та навчився їх програмно реалізовувати мовою С++.

У ході виконання роботи потрібно порівняти різницю структурного програмування. У структурному підході немає інкапсуляції, і доступ до даних здійснюється безпосередньо, що робить код простішим, але менш гнучким у великих проєктах, також немає механізму для приховування даних або чіткої організації коду навколо "об'єктів".