Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

Дисципліна: Об’єктно-орієнтоване програмування

**Лабораторна робота №1**

**Тема:** **«ОСНОВНІ ПОНЯТТЯ ООП. КЛАСИ ТА ОБ’ЄКТИ. ФУНКЦІЇ ДОСТУПУ. ВКАЗІВНИК THIS»**

|  |
| --- |
| Виконав: ст. гр. КI-23 |
| Яровець Т.О. |
| Перевірив викладач:  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:**

Завдання 1

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

Приватні поля класу:

• name (ім'я студента)

• age (вік студента)

• major (спеціальність студента)

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.

2. Створіть об'єкт класу "Student".

3. Задайте значення полів об'єкта за допомогою відповідних методів.

4. Виведіть інформацію про студента на екран, використовуючи методи для отримання значень полів.

5. Реалізувати програму за допомогою роздільної компіляції. У вашому рішенні можуть бути додаткові методи та поля, якщо ви вважаєте їх необхідними.

Завдання 2

Реалізувати вище наведену задачу за допомогою структурного програмування. У висновку описати різницю цих методів.

Лістинг завдання 1(task1):

main.cpp:

#include <iostream>

#include <student.h>

using namespace std;

int main()

{

Student Person;

Person.setName("Vadim");

Person.setAge(17);

Person.setMajor("Computer engineering");

cout << "Person name: " << Person.getName() << endl;

cout << "Person age: " << Person.getAge() << endl;

cout << "Person major: " << Person.getMajor() << endl;

return 0;

}

student.cpp:

#include "student.h"

Student::Student() {}

void Student::setName(string newName)

{

name = newName;

}

string Student::getName()

{

return name;

}

void Student::setAge(int newAge)

{

age = newAge;

}

int Student::getAge()

{

return age;

}

void Student::setMajor(string newMajor)

{

major = newMajor;

}

string Student::getMajor()

{

return major;

}

student.h:

#ifndef STUDENT\_H

#define STUDENT\_H

#include <string>

using namespace std;

class Student

{

public:

Student();

void setName(string newName);

string getName();

void setAge(int newAge);

int getAge();

void setMajor(string newMajor);

string getMajor();

private:

string name;

int age;

string major;

};

#endif // STUDENT\_H

Лістинг завдання 2(task2):

main.cpp:

#include <iostream>

#include <student.h>

using namespace std;

int main()

{

Student Person;

Person.name = "Vadim";

Person.age = 17;

Person.major = "Computer engineering";

cout << "Person name: " << Person.name << endl;

cout << "Person age: " << Person.age << endl;

cout << "Person major: " << Person.major << endl;

return 0;

}

student.cpp:

#include "student.h"

void setName(Student &student, const string &newName)

{

student.name = newName;

}

string getName(const Student &student)

{

return student.name;

}

void setAge(Student &student, int newAge)

{

student.age = newAge;

}

int getAge(const Student &student)

{

return student.age;

}

void setMajor(Student &student, string newMajor)

{

student.major = newMajor;

}

string getMajor(const Student &student)

{

return student.major;

}

student.h:

#ifndef STUDENT\_H

#define STUDENT\_H

#include <string>

using namespace std;

struct Student{

string name;

int age;

string major;

};

void setName(Student &student, string &newName);

string getName(Student &student);

void setAge(Student &student, int newAge);

int getAge(Student &student);

void setMajor(Student &student, string &newMajor);

string getMajor(Student &student);

#endif // STUDENT\_H

**Висновок:**

В класі всі дані приватні, тому доступ до них здійснюється через публічні методи, для того, щоб захистити від зміни даних. Але в структурі дані зазвичай публічні, тому функції можуть змінювати їх, що може призвести небажані зміни в роботі програми, в структурі також відсутня спадковість та поліморфізм на відміну від класів, в яких методи можуть бути перевизначені у підкласах.