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. КБ-23 |
| Захарченко А.О. |
| Перевірив: викладач  Козірова Наталія Леонідівна |
|  |

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 3:**

**Завдання 1:**

1. Створіть клас "Person" (людина), який має наступні властивості та функціональність:

Приватні поля класу:

* name (ім'я людини)
* age (вік людини)
* address (адреса людини)

Публічні методи класу:

* Метод setName(), який дозволяє задати ім'я людини.
* Метод getName(), який повертає ім'я людини.
* Метод setAge(), який дозволяє задати вік людини.
* Метод getAge(), який повертає вік людини.
* Метод setAddress(), який дозволяє задати адресу людини.
* Метод getAddress(), який повертає адресу людини.

2. Створіть об'єкт класу "Person".

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

4. Виведіть інформацію про людину 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.

**Реалізація завдання 1:**

**Файл Person.cpp:**

#include "Person.h"

using namespace std;

void Person::setname(const string name) {

this->name = name;

}

void Person::setage(int age) {

this->age=age;

}

void Person::setaddress(const string address) {

this->address=address;

}

string Person::getname() {

return name;

}

int Person::getage() {

return age;

}

string Person::getaddress() {

return address;

}

**Файл Person.h:**

#pragma once

#ifndef PERSON\_H

#define PERSON\_H

#include <string>

using namespace std;

class Person

{

private:

string name;

int age;

string address;

public:

string getname();

int getage();

string getaddress();

void setname(string name);

void setage(int age);

void setaddress(string address);

};

#endif

**Файл Task-1.cpp:**

#include <iostream>

#include "Person.h"

using namespace std;

int main() {

Person person;

string name;

int age;

string address;

cout << "Name: " << endl;

cin >> name;

person.setname(name);

cout << "Age: " << endl;

cin >> age;

person.setage(age);

cout << "Address: " << endl;

cin >> address;

person.setaddress("address");

cout << "Name: " << person.getname() << endl;

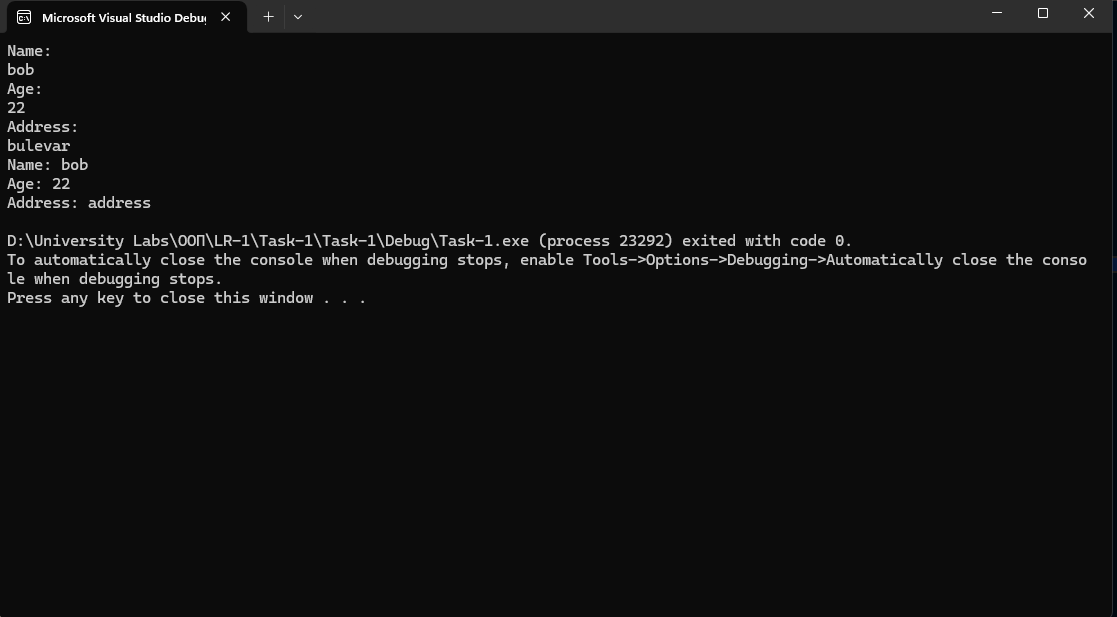
cout << "Age: " << person.getage() << endl;

cout << "Address: " << person.getaddress() << endl;

return 0;

}

**Результат виконання:**



**Реалізація завдання 2:**

#include <iostream>

#include <string>

using namespace std;

struct Person {

string Name, Address;

int Age;

void setName(string name) {

this->Name=name;

}

void setAge(int age) {

this->Age = age;

}

void setAddress(string address) {

this->Address = address;

}

string getName() {

return Name;

}

int getAge() {

return Age;

}

string getAddress() {

return Address;

}

};

int main() {

Person pers;

string Name;

int Age;

string Address;

cout << "Name: ";

cin >> Name;

pers.setName(Name);

cout << "Age: ";

cin >> Age;

pers.setAge(Age);

cout << "Address: ";

cin >> Address;

pers.setAddress(Address);

cout << "Name: " << pers.getName() << endl;

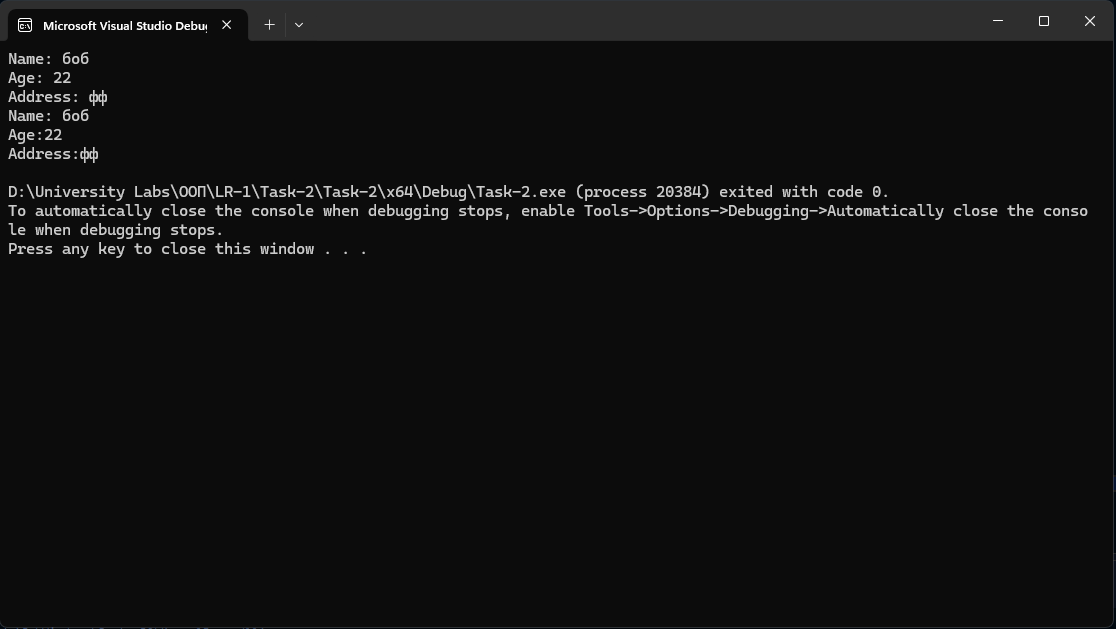
cout<<"Age:" << pers.getAge()<<endl;

cout << "Address:" << pers.getAddress() << endl;

return 0;

}

**Результат:**



**Висновок:** отже виконавши ЛР1 можу сказати, що покращив навички набирання коду на мові програмування С++, а також поглибив розуміння класів та нагадав саму структури структур. Щодо порівняння двох задач та виконання поставлених умов можна сказати що структурне програмування підходить для невеликих задач, а ось класи більш підходять для великих проєктів.