Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ №1

з дисципліни «Об’єктно-орієнтоване програмування»

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

|  |
| --- |
| Виконав: ст. гр. КН-23 |
| Гребенюк Д.О. |
| Перевірив: асистент  Козірова Н.Л. |
|  |

Кропивницький

2024

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

**Варіант 4**

**Завдання 1**

1. Створіть клас "Employee" (співробітник), який має наступні властивості

та функціональність:

Приватні поля класу:

• name (ім'я співробітника)

• id (ідентифікатор співробітника)

• salary (заробітна плата співробітника)

Публічні методи класу:

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

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

• Метод setId(), який дозволяє задати ідентифікатор співробітника.

• Метод getId(), який повертає ідентифікатор співробітника.

• Метод setSalary(), який дозволяє задати заробітну плату співробітника.

• Метод getSalary(), який повертає заробітну плату співробітника.

2. Створіть об'єкт класу "Employee".

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

4. Виведіть інформацію про співробітника 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.

**Завдання 1**

*Лістинг employee.h*

#ifndef EMPLOYEE\_H

#define EMPLOYEE\_H

#include <string>

class Employee

{

public:

Employee();

std::string getName();

void setName(std::string newName);

int getId();

void setId(int newId);

int getSalary();

void setSalary(int newSalary);

private:

std::string name;

int id;

int salary;

};

#endif // EMPLOYEE\_H

*Лістинг employee.cpp*

#include "employee.h"

Employee::Employee() {

name = "";

id = -1;

salary = -1;

}

std::string Employee::getName()

{

return name;

}

void Employee::setName(std::string newName)

{

name = newName;

}

int Employee::getId()

{

return id;

}

void Employee::setId(int newId)

{

id = newId;

}

int Employee::getSalary()

{

return salary;

}

void Employee::setSalary(int newSalary)

{

salary = newSalary;

}

*Лістинг main.cpp*

#include "employee.h"

#include <iostream>

int main()

{

Employee alex;

alex.setId(1);

alex.setName("Alex");

alex.setSalary(10000000);

std::cout << "Id: " << alex.getId() << std::endl;

std::cout << "Name: " << alex.getName() << std::endl;

std::cout << "Salary: " << alex.getSalary() << std::endl;

return 0;

}

**Завдання 2**

*Лістинг employee.h*

#ifndef EMPLOYEE\_H

#define EMPLOYEE\_H

#include <string>

struct Employee {

std::string name = "";

int id = -1;

int salary = -1;

};

void setName(Employee &object, std::string newName);

std::string getName(Employee &object);

void setId(Employee &object, int newId);

int getId(Employee &object);

void setSalary(Employee &object, int newSalary);

int getSalary(Employee &object);

#endif // EMPLOYEE\_H

*Лістинг employee.cpp*

#include "employee.h"

void setName(Employee &object, std::string newName) {

object.name = newName;

}

std::string getName(Employee &object) {

return object.name;

}

void setId(Employee &object, int newId) {

object.id = newId;

}

int getId(Employee &object) {

return object.id;

}

void setSalary(Employee &object, int newSalary) {

object.salary = newSalary;

}

int getSalary(Employee &object) {

return object.salary;

}

*Лістинг main.cpp*

#include <iostream>

#include "employee.h"

int main()

{

Employee bob;

setId(bob, 2);

setName(bob, "Bob");

setSalary(bob, 99999);

std::cout << "Id: " << getId(bob) << std::endl;

std::cout << "Name: " << getName(bob) << std::endl;

std::cout << "Salary: " << getSalary(bob) << std::endl;

return 0;

}

**Висновок:**

В ході виконання лабораторної роботи я ознайомився з основними поняттями об’єктно-орієнтованого програмування, такими як класи та об'єкти, а також навчився використовувати функції доступу до приватних полів через сеттери та геттери. Під час роботи над завданням №2 я займався порівнянням структурного та об’єктно-орієнтованого підходу програмування. Основна відмінність між структурним та об’єктно орієнтованим програмуванням - у тому, що структурне програмування дає змогу розробити програму за допомогою набору модулів або функцій. Тоді як об’єктно орієнтоване -побудувати програму за допомогою набору об’єктів та їхньої взаємодії.