МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №2**

з дисципліни «Об’єктно-орієнтоване програмування»

на тему: «Робота з об’єктами, використання покажчика this»

Варіант №5

**Виконав:**

студент гр. БС-71

Воробйов І.О.

**Перевірив:**

асистент каф. БМК

Рисін С.В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2019

**🞏 Практичне заняття без зауважень**

**🞏 Практичне заняття має зауваження:**

**🞏 несвоєчасний захист**

**🞏 присутні зауваження до UML діаграми:**

**🞏 діаграма класу не відповідає коду**

**🞏 виконані не за стандартом:**

**🞏 атрибути**

**🞏 відношення**

**🞏 потужність**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 присутні зауваження до коду:**

**🞏 задача завдання вирішена хибно**

**🞏 код програми не компілюється**

**🞏 хибно задані специфікатори доступу**

**🞏 помилки у визначенні конструкторів / деструкторів**

**🞏 відсутні списки ініціалізації в конструкторах**

**🞏 константні методи**

**🞏 використано глобальні змінні**

**🞏 статичні змінні при роботі з масивами**

**🞏 оформлення коду**

**🞏 присутні зайві символи «{» та «}»**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 невірні відповіді на запитання:**

**🞏 №1 🞏 №2 🞏 №3 🞏 №4 🞏 №5**

**🞏 №6 🞏 №7 🞏 №8 🞏 №9 🞏 №10**

**🞏 маються інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Завдання:**

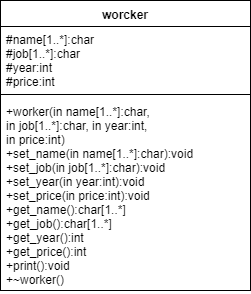
1. Вивчити теоретичні основи роботи з об’єктами та використання покажчика this.
2. Скористатися розробленим в попередній лабораторій роботі класом для розширення його функціональності (нові методи класу мають працювати з покажчиком this).
3. Визначення класів та їх реалізації мають бути розташовані у файлах \*.h та \*.cpp відповідно.
4. Відповідно до свого варіанту написати програму, яка містить клас для роботи із запитами користувача (методи класу дозволяють відповідно до вибору користувача вивести на екран задані в завданні списки) та інкапсулює в собі масив об’єктів розробленого в попередній лабораторій роботі класу:

***Створити масив об’єктів та вивести на екран:***

* ***список працівників, стаж роботи яких перевищує задане число років;***
* ***список працівників, зарплата яких більше заданої;***
* ***список працівників, які займають задану посаду.***

1. Скласти і захистити звіт.

**UML діаграма**



**Код програми:**

**worcker.h:**

#pragma once

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <cstring>

#include <iostream>

using namespace std;

class worcker

{

private:

char \*name;

char \*job;

int year;

int price;

public:

worcker() {};

worcker(const char \*, const char \*, int, int);

void set\_name(const char \*);

void set\_job(const char \*);

void set\_year(int);

void set\_price(int);

const char \* get\_name();

const char \* get\_job();

int get\_year();

int get\_price();

void print();

~worcker();

};

**worcker.cpp:**

#include "worcker.h"

worcker::worcker(const char \* Name, const char \*Job, int Year, int Price) :year(Year), price(Price)

{

name = new char[50];

strcpy(name, Name);

job = new char[50];

strcpy(job, Job);

}

void worcker::set\_name(const char \*a)

{

if (name != NULL)

delete[]name;

name = new char[20];

strcpy\_s(name, 20, a);

}

void worcker::set\_job(const char \*a)

{

if (job != NULL)

delete[]job;

job = new char[20];

strcpy\_s(job, 20, a);

}

void worcker::set\_year(int a)

{

year = a;

}

void worcker::set\_price(int a)

{

price = a;

}

const char \* worcker::get\_name()

{

return this->name;

}

const char \* worcker::get\_job()

{

return this->job;

}

int worcker::get\_year()

{

return this->year;

}

int worcker::get\_price()

{

return this->price;

}

worcker::~worcker()

{

delete[]name;

delete[]job;

}

void worcker::print()

{

cout << "Name = " << name << "\nJob = " << job << "\nYear = " << year << "\nPrice = " << price << endl;

}

**Source.cpp:**

#include "worcker.h"

int main()

{

worcker \*\*man = new worcker\*[10];

for (size\_t i = 0; i < 10; i++)

{

char \*name = new char[50];

char \*job = new char[50];

int year, price;

cout << "Input name : "; cin >> name;

cout << "Input job : "; cin >> job;

cout << "Input year : "; cin >> year;

cout << "Input price : "; cin >> price;

man[i] = new worcker(name, job, year, price);

}

int price;

cout << "Input price = "; cin >> price;

for (size\_t i = 0; i < 10; i++)

if (man[i]->get\_price() >= price)

man[i]->print();

double experience;

cout << "Input experience = "; cin >> experience;

for (size\_t i = 0; i < 10; i++)

if (2019-man[i]->get\_year() >= experience)

man[i]->print();

char \*job = new char[50];

cout << "Input job = "; cin >> job;

for (size\_t i = 0; i < 10; i++)

if (man[i]->get\_job() == job)

man[i]->print();

for (size\_t i = 0; i < 10; i++)

delete man[i];

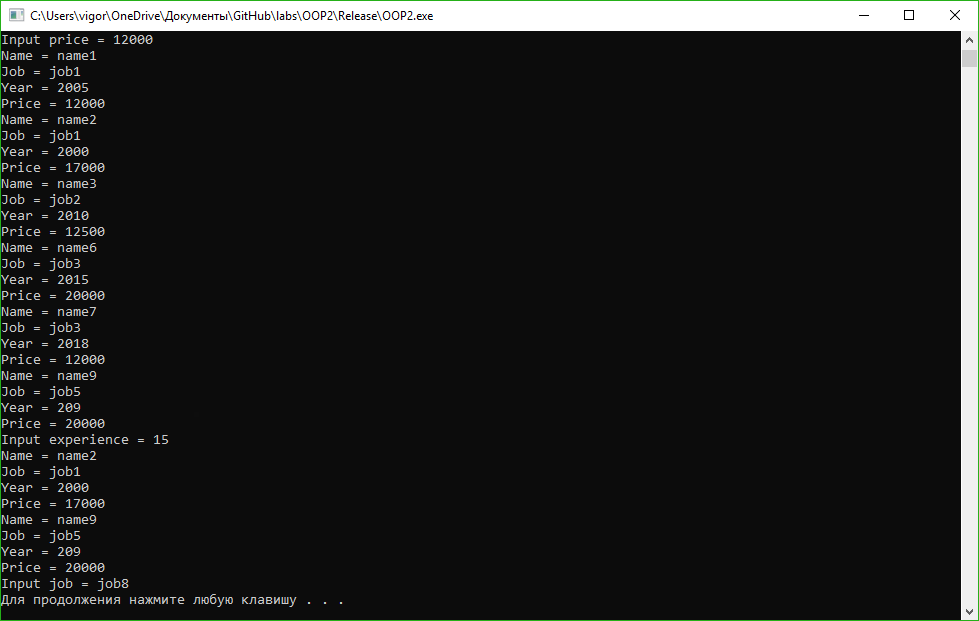
delete[]man;

system("pause");

return 0;

}

**Результат роботи програми:**



**Контрольні запитання**

1. Що таке покажчик this?

Указатель this указывает на объект для которого выполняется функция-член. Он доступный только в нестатических функциях-членов class, struct, union.

1. Навести приклад, в якому не можна організувати код без використання покажчика this.

void bus::setNumberPlace(int number\_place)

{

this->number\_place = number\_place;

}

1. Два способи ініціалізації об’єктів у програмі.

Ініціалізувати об’єкт можна за допомогою конструктора з параметром та без параметра.

1. Як ініціалізувати масив об’єктів?

class ob[3] = {

class(1, 2),

class(3, 4 ),

class(5, 6)

};

1. Чому не можна ініціалізувати поле, що є масивом, під час визначення класу?

Так как массив не является статическим(константным) полем и память под объекты еще не выделена.

1. Перелічити випадки, коли один клас використовує інший.

Отношение зависимости, зависимость – отношение между двумя классами, при котором изменение одного класса (поставщика) может затронуть другой класс (клиент) или предоставить ему необходимую информацию. Отношение ассоциации (ассоциация, association) – описывает связи между экземплярами классов (объектами) - в отличие от зависимости, которая относится к классу в целом. В ассоциации проставляется множественность участия экземпляров в связи (один или много). Отношение агрегации (агрегация, aggregation) (рисунок 5) – один из классов (агрегат) состоит из других классов или его характеристиками являются другие классы (отношение часть/ целое, part of). Композиция (composition) – усиленная форма агрегации, в которой агрегат, называемый композитом, несёт полную ответственность за создание и уничтожение своих частей, т.е. самостоятельно классы - части композита существовать не могут.

1. Які є види відношень між класами?

Мы можем выделить несколько основных отношений: наследование, реализация, ассоциация, композиция и агрегация.

1. Які види відношень між класами визначені в мові моделювання UML?

Наследование, реализация, ассоциация, композиция и агрегация, композиции.

1. Який тип відношення вставлено між класами в даній лабораторній роботі?

Агрегация.

1. Що означає оператор(::)?

Оператор разрешения области :: используется для идентификации и устранения неоднозначности идентификаторов, которые используются в разных областях.