МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №1**

з дисципліни «Об’єктно-орієнтоване програмування»

на тему: «Реалізація класів, конструктори та деструктори»

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:**

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

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

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

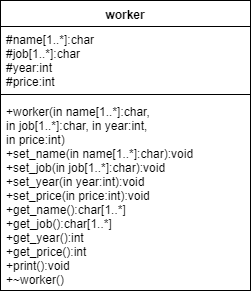
**Завдання:**

1. Вивчити теоретичні відомості щодо створення класів та об’єктів мовою програмування C++.
2. Відповідно до свого варіанту розробити клас та відповідну діаграму класів в нотації UML для описаних в завданні характерних даних, визначити конструктор користувача, за допомогою якого можна ініціалізувати всі поля визначеного класу, а також методи для зміни значень кожного із полів:

***Створити клас, що характеризує робітника: прізвище та ініціали, посада, рік вступу на роботу, зарплата.***

1. Розробити програмний додаток, в якому необхідно створити об’єкт, що є екземпляром визначеного в завданні класу, та ініціалізувати його поля за допомогою створеного конструктора користувача, вивести інформацію про створений об’єкт на екран, змінити значення полів об’єкта за допомогою викликів відповідних методів класу, після чого знову вивести інформацію про модифікований об’єкт на екран, видалити створений об’єкт.
2. Скласти і захистити звіт.

**UML діаграма**



**Код програми:**

**worker.h:**

#pragma once

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <cstring>

#include <iostream>

using namespace std;

class worker

{

private:

char \*name;

char \*job;

int year;

int price;

public:

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

~worker();

};

**worker.cpp:**

#include "worker.h"

worker::worker(const char \* Name, int sizeofname, const char \*Job, int sizeofjob, int Year, int Price) :year(Year), price(Price)

{

name = new char[sizeofname];

strcpy(name, Name);

job = new char[sizeofjob];

strcpy(job, Job);

}

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

{

if (name != NULL)

delete[]name;

name = new char[20];

strcpy\_s(name,20, a);

}

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

{

if (job != NULL)

delete[]job;

job = new char[20];

strcpy\_s(job, 20, a);

}

void worker::set\_year(int a)

{

year = a;

}

void worker::set\_price(int a)

{

price = a;

}

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

{

return name;

}

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

{

return job;

}

int worker::get\_year()

{

return year;

}

int worker::get\_price()

{

return price;

}

worker::~worker()

{

delete[]name;

delete[]job;

}

void worker::print()

{

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

}

**Source.cpp:**

#include "worker.h"

#include <conio.h>

int main()

{

worker man = worker("name", 4, "job", 3, 2019, 10000);

man.print();

char \*name = new char[20];

char \*job = new char[20];

int year, price;

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

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

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

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

man.set\_name(name);

man.set\_job(job);

man.set\_year(year);

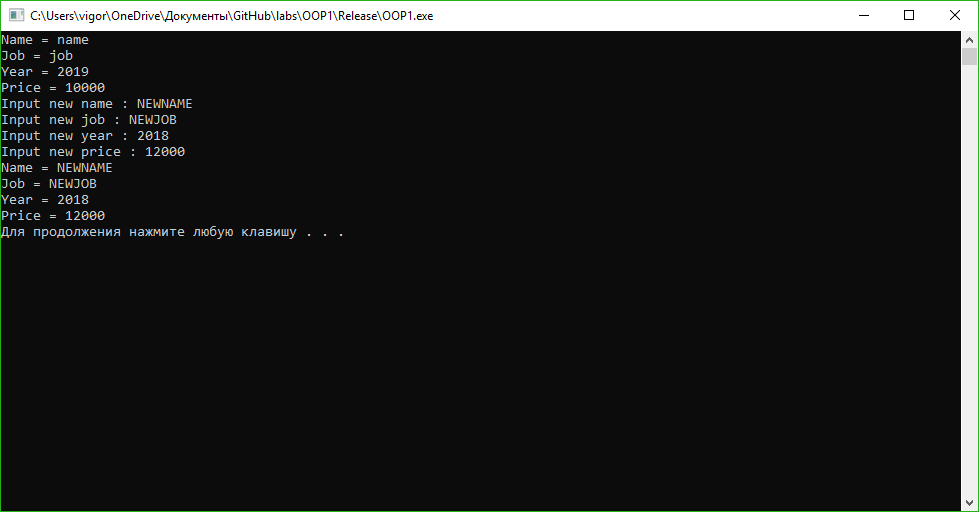
man.set\_price(price);

man.print();

system("pause");

return 0;

}

**Результат роботи програми:**

**Контрольні запитання**

1. Що таке клас, навіщо він потрібен?

Клас – це абстрактний тип даних. Клас визначає новий тип даних, який задає формат об’єкта і принципи його обробки.

1. Що таке об’єкт?

Объект (экземпляр) – это единица данных отражающая сущность реального мира с точки зрения программирования.

1. Як пов’язані між собою класи і об’єкти в програмі?

В программе объект является переменной тип которой класс. Отдельный представитель класса, имеющий конкретное состояние и поведение, полностью определяемое классом.

1. Що таке інкапсуляція?

Інкапсуляція - механізм, який поєднує дані та методи, що обробляють ці дані. Коли і методи і данні об’єднуються таким чином — створюється об’єкт.

1. За рахунок чого реалізується захист від несанкціонованного досупу до данних?

За допомогою специфікаторов доступу private и protected. Private визначає закриті члени класу, що доступні лише усередині інших членів того самого класу. Protected визначає захищені члени класу, що доступні як для членів того ж самого класу, так і для членів похідних класів.

1. Чим відрізняються поля від змінних?

Поля – это переменные, связаная с классом или объектом. Доступ к полям осуществляется по их имени. Обычно тип данных каждого поля задаётся в описании класса, членом которого является поле.

1. Що визначають методи класу?

Методы определяет поведение объекта класса. Поведение – это действие которое можно выполнять с классом.

1. Перелічити, чим можуть бути поля класу?

Все, кроме объекта этого класса прямо или опосредовано.

1. Що таке конструктер, навіщо він потрібен?

Конструктор (от слова construct — создавать) – это специальный метод класса, который предназначен для инициализации элементов класса некоторыми начальными значениями.

1. Що таке деструктор, навіщо він потрібен?

Деструктор (от слова destruct — разрушать) — специальный метод класса, который служит для уничтожения элементов класса. Чаще всего его используют тогда, когда в конструкторе,при создании объекта класса, динамически был выделен участок памяти и необходимо эту память очистить, если эти значения уже не нужны для дальнейшей работы программы.