 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №3**

з дисципліни «Об’єктно-орієнтоване програмування»

на тему: «Успадкування. Поліморфізм»

Варіант №15

**Виконав:**

студент гр. БС-71

Орлівський С. П.

**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:**

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

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

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

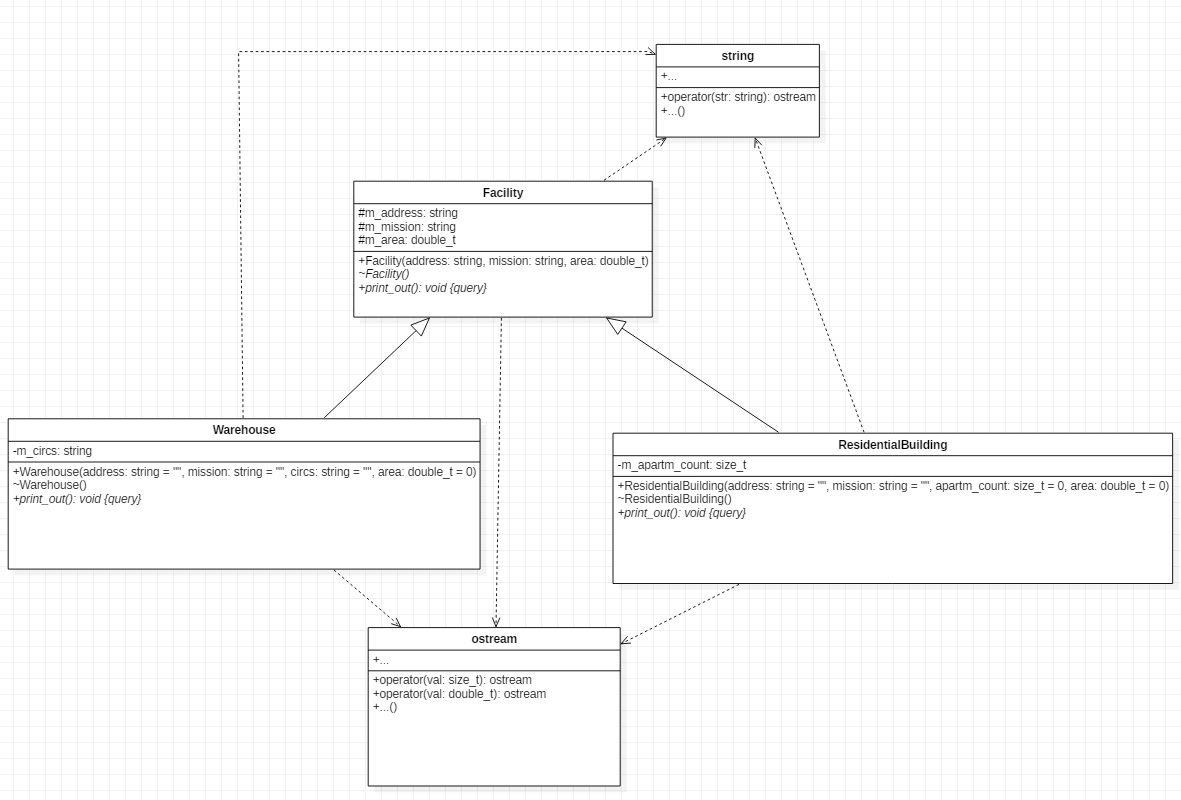
**Завдання:**

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 квартиру жилого будинку: адреса, поверх, площа, кількість кімнат, площа.***

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:**

**facility.h:**

#pragma once

#include <string>

#include <iostream>

#include <cmath>

class **Facility**

{

public:

**Facility**(std::string address = "", std::string mission = "", double\_t area = 0);

virtual void ***print\_out***() const = 0;

virtual ~***Facility***() = 0;

protected:

std::string m\_address, m\_mission;

double\_t m\_area;

};

**facility.cpp:**

#include "facility.h"

Facility::**Facility**(std::string address, std::string mission, double\_t area)

: m\_address(address), m\_mission(mission), m\_area(area)

{

}

Facility::~***Facility***()

{

}

void Facility::***print\_out***() const

{

std::cout << "Address: " << m\_address << std::endl

<< "Mission: " << m\_mission << std::endl

<< "Area: " << m\_area << std::endl;

}

**warehouse.h:**

#pragma once

#include "facility.h"

class **Warehouse** : public Facility

{

public:

**Warehouse**(std::string address = "", std::string mission = "",

std::string circs = "", double\_t area = 0);

void ***print\_out***() const;

~***Warehouse***();

private:

std::string m\_circs;

};

**warehouse.cpp:**

#include "warehouse.h"

Warehouse::**Warehouse**(std::string address, std::string mission, std::string circs, double\_t area)

:Facility (address, mission, area), m\_circs(circs)

{

}

Warehouse::~***Warehouse***()

{

}

void Warehouse::***print\_out***() const

{

std::cout << "\n\n" << "Class: Warehouse\n";

Facility::print\_out();

std::cout << "Circumstances: " << m\_circs << "\n\n";

}

**residentialbuilding.h:**

#pragma once

#include "facility.h"

class **ResidentialBuilding** : public Facility

{

public:

**ResidentialBuilding**(std::string address = "", std::string mission = "",

size\_t apartm\_count = 0, double\_t area = 0);

~***ResidentialBuilding***();

void ***print\_out***() const;

private:

size\_t m\_apartm\_count;

};

**residentialbuilding.cpp:**

#include "residentialbuilding.h"

ResidentialBuilding::**ResidentialBuilding**(std::string address, std::string mission, size\_t apartm\_count, double\_t area)

: Facility(address, mission, area), m\_apartm\_count(apartm\_count)

{

}

ResidentialBuilding::~***ResidentialBuilding***()

{

}

void ResidentialBuilding::***print\_out***() const

{

std::cout << "\n\n" << "Class: ResidentialBuilding\n";

Facility::print\_out();

std::cout << "Number of apartments: " << m\_apartm\_count << "\n\n";

}

**main.cpp:**

#include <iostream>

#include "facility.h"

#include "warehouse.h"

#include "residentialbuilding.h"

int **main**()

{

ResidentialBuilding rb("351234513", "wqereqwrt", 1124, (double\_t)234.345);

Warehouse wh("56783", "edsfgkjfdsj", "good", (double\_t)345.4);

Facility \*facility\_ptr;

facility\_ptr = &rb;

facility\_ptr->*print\_out*();

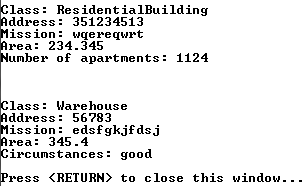
facility\_ptr = &wh;

facility\_ptr->*print\_out*();

return 0;

}

**Результат роботи програми:**

****

**Контрольні запитання**

* 1. Що означає поняття успадкування?

Поняття успадкування означає, що один клас може успадкувати усі методи та поля (поведінку) класу від якого він наслідується (супер/базового/батьківського класу)

* 1. Який клас називається базовим, а який є класом нащадком?

Базовий клас – клас від якого унаслідуваний хоча б один клас. Нащадок – клас, що наслідується (є похідним) від базового класу.

* 1. Скільки базових класів може мати похідний клас?

Від одного до безлічи.

* 1. Чи може похідний клас бути базовим?

Так, базовим для класів, котрі наслідуються від нього.

* 1. Як змінюється доступ до елемегтів базового класу при спадкуванні з різними специфікаторами доступу?

На такий випадок я маю табличку, в котрій все набагато зрозуміліше виглядає, ніж, якщо я почну словесно усе розписувати:



* 1. У чому різниця між простим та множинним наслідуванням?

При множинному наслідуванні клас копіює поведінку усіх класів від яких він наслідується (усі поля та методи). При одинармому наслідуванні відбувається теж саме з поправкою, що базовий клас всього один.

Nf

7. Чи можна з класу-нащадка одержати доступ до private частини базового класу, якщо специфікатор доступу при спадкуванні private?

До приватної частини базового класу не можна доступитися з класів нащадків при жодному специфікаторі наслідування.

8. Чи успадковуються конструктори та деструктори?

Ні.

9. Що таке віртуальна функція?

Віртуальна функція – це така функція базового класу, що може бути перевизначена у класах нащадках. При цьому така функція викликається під час run time. При виклику такої функції через вказівник на базовий клас, що вказує на клас-нащадок, буде викликана саме перевизначена функція класу нащадка, якщо така є (на відміну від не віртуальних функцій).

10.Чи можна віртуальну функцію визначити як static, відповідь пояснити.

Ні, не можна. Адже віртуальна функція працює з конкретним об’єктом того класу, що її перевизначив, а staic функція може бути викликана взагалі без об’єктів.