 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 №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:**

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

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

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

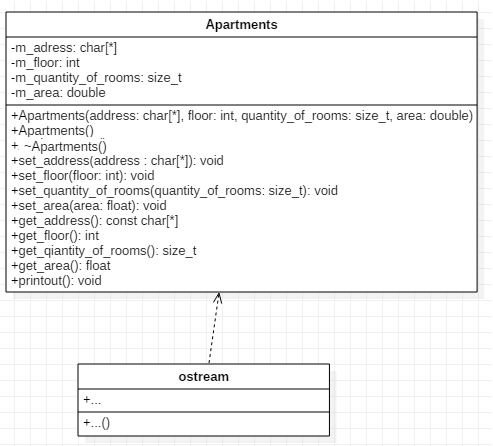
**Завдання:**

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:**

**Source.h:**

class Apartments

{

private:

char \*m\_address;

int m\_floor;

size\_t m\_quantity\_of\_rooms;

double m\_area;

public:

Apartments(const char \*address, const int floor, const size\_t quantity\_of\_rooms,

const double area);

Apartments();

void set\_address(const char \*address);

void set\_floor(const int floor);

void set\_quantity\_of\_rooms(const size\_t quantity\_of\_rooms);

void set\_area(const double area);

const char \* get\_address() const;

int get\_floor() const;

double get\_area() const;

size\_t get\_quantity\_of\_rooms() const;

void printout() const;

~Apartments();

};

**Source.cpp:**

Apartments::Apartments(const char \*address, const int floor, const size\_t quantity\_of\_rooms,

const double area):m\_address(new char [strlen(address)+1]),

m\_floor(floor), m\_quantity\_of\_rooms(quantity\_of\_rooms), m\_area(area)

{

strcpy(m\_address, address);

}

Apartments::Apartments() :m\_address(nullptr), m\_floor(0), m\_quantity\_of\_rooms(0),

m\_area(0) {}

Apartments::~Apartments()

{

delete[] m\_address;

}

void Apartments::set\_address(const char \*address)

{

delete[] m\_address;

m\_address = new char[strlen(address) + 1];

strcpy(m\_address, address);

}

void Apartments::set\_floor(const int floor)

{

m\_floor = floor;

}

void Apartments::set\_quantity\_of\_rooms(const size\_t quantity\_of\_rooms)

{

m\_quantity\_of\_rooms = quantity\_of\_rooms;

}

void Apartments::set\_area(const double area)

{

m\_area = area;

}

const char \*Apartments::get\_address() const

{

return (!m\_address) ? "" : m\_address;

}

int Apartments::get\_floor() const

{

return m\_floor;

}

double Apartments::get\_area() const

{

return m\_area;

}

size\_t Apartments::get\_quantity\_of\_rooms() const

{

return m\_quantity\_of\_rooms;

}

void Apartments::printout() const

{

std::cout << "Address: " << this->get\_address() << std::endl

<< "Floor: " << this->get\_floor() << std::endl

<< "Quantity of rooms: " << this->get\_quantity\_of\_rooms() << std::endl

<< "Area: " << this->get\_area() << std::endl;

}

**main.cpp:**

int main()

{

Apartments \*flat = new Apartments("st A, 69-7", 7, 7, 777);

flat->printout();

flat->set\_address("ts 7-96");

flat->set\_floor(14);

flat->set\_quantity\_of\_rooms(14);

flat->set\_area(141414);

std::cout << "New data: \n";

flat->printout();

delete flat;

flat = new Apartments("st A, 69-7", 7, 7, 777);

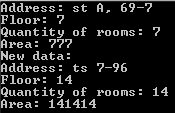
flat->printout();

getchar();

return 0;

}

**Результат роботи програми:**

****

**Контрольні запитання**

* 1. Що таке клас, навіщо він потрібен?

Класс – це абстрактний тип данних користувача, що описується самим користувачем. Потрібен для більш зручного представлення та вирішення задачі програмування.

* 1. Що таке об’єкт?

Об’єкт – це сутність, що має поля та методи, котрі за рахунок принципу абстрактноті відображають поведінку та характеристики реального об’єкту. За рахунок цього ця сутність також представляє об’єкт реального світу. Хоча об’єкти також можуть представляти і чисто абстрактні концепти.

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

Класи – це інструкції щодо внутрішньої структури кожного об’єкту (шаблон об’єкту). Кожний об’єкт має тип того класу, по інструкціям якого буде створена його структура і називається екземпляром цього класу.

* 1. Що таке інкапсуляція?

Інкапсуляція – це об’єднання характеристик описуємого об’єкту як єдиного цілого. На основі інкапсуляції створюються структурні секції з різним рівнем доступу.

* 1. За рахунок чого реалізується захист від несанкціонованого доступу до даних?

За рахунок модифікаторів безпеки prіvate та protected. Поле, що позначено модифікатором private видиме та доступне лише у середені класу. А protected – у самому класі та класах нащадках.

1. Чим відрізняються поля від змінних?

Поля це змінні, що знаходяться всередені класу. Тобто усяке поле – це змінне, але не всяка змінна це поле.

1. Що визначають методи класу?

Методи классу визначають функціональні властивості об’єкту класу (його функції), тобто визначають поведінку його екземплярів.

1. Чим можуть бути поля класу?

Поля класу можуть буди будь-якого типу крім voіd, типу самого класу та типу данних іншого класу в полях якого зберігаються об’єкти, що містять тип данних початкового класу.

Nf

1. Що таке конструктор, навіщо він потрібен?

Конструктор класу – це метод класу, що автоматично викликається при створенні об’єкту цьго классу та виділяє пам’ять під всі поля об’єкту, тобто створює його.

1. Що таке деструктор, навіщо він потрібен?

Деструктор класу – це метод класу, що автоматично викликається при видаленні об’єкту та видаляє його поля з оперативної пам’яті.