Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра “Електронних обчислювальних машин”



**Звіт з лабораторної роботи №4**

**Програмування, частина 2**

**(Об’єктно - орієнтоване програмування)**

на тему:

“Класи та об’єкти”

Варіант - 25

**Виконала:**

ст. гр. КІ-15

Романішина А. Ю.

**Перевірив:**

асист. каф. ЕОМ

Козак Н. Б.

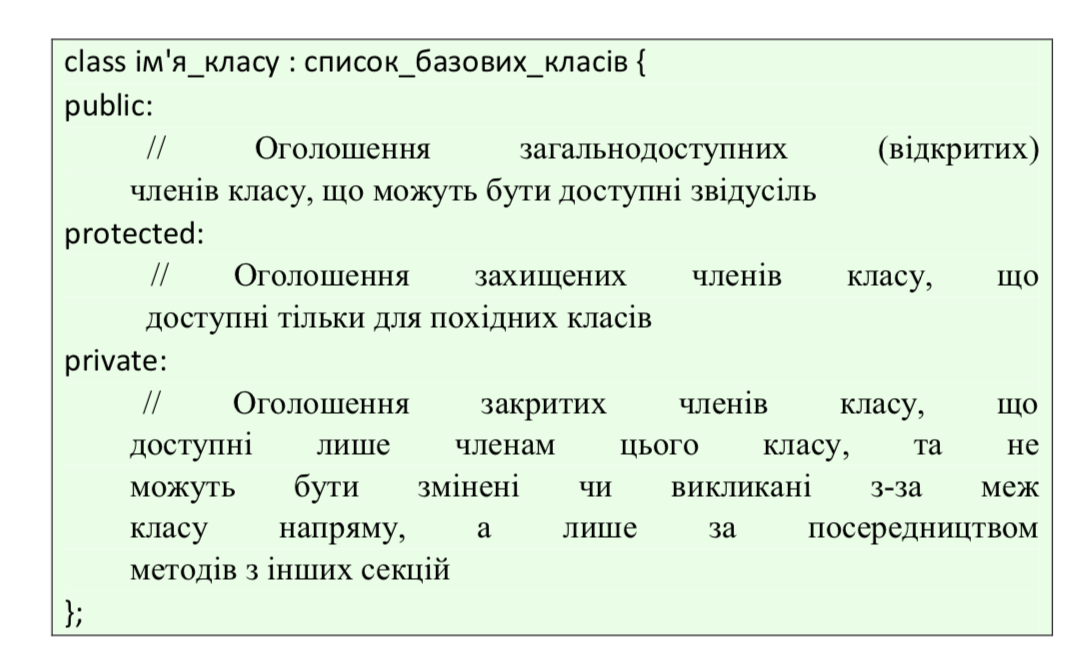
Львів – 2020

**Мета:** познайомитися із класами та об‘єктами.

**Теоретичні відомості:**

*Визначення класу:*

Основна відмінність будь-якої об‘єктно-орієнтованої мови програмування від інших не об‘єктно-орієнтованих мов програмування є можливість створення нових визначених користувачем типів, що називаються класами. Клас – це визначений користувачем тип з даними-елементами (властивостями) та функціями-елементами (методами), що являються членами класу. Він зазвичай описує певну абстракцію якоїсь сутності реального світу з її властивостями і можливими діями. Змінна типу клас називається об‘єктом. Об‘єкт – це вже не абстракція, а елемент реального світу, який може володіти певними характеристиками, які задаються властивостями в класі, та діяти згідно операцій заданих методами у класі. Оголошення класу в мові С++ має наступний синтаксис (не обов‘язково щоб клас мав всі секції чи спадкував базовий клас):



Члени класу (властивості і методи), оголошені після ключового слова publіc стають відкритими членами. Це означає, що вони доступні:

• усім іншим членам класу;  
• дружнім конструкціям (класам, методам, функціям);  
• членам похідних класів;  
• з-під об‘єктів класу після створення об‘єктів; після створення об‘єкту

класу до його членів можна звертатися лише якщо вони є загальнодоступними.

Члени класу, оголошені після ключового слова prіvate, стають закритими членами. Вони доступні:

• для інших членів того ж класу; • друзям класу.

Якщо в класі не оголошено секцій, але визначено методи чи властивості, то вони вважатимуться такими, що оголошені у секції prіvate. Для модифікації властивостей, що оголошені в секції private створюються спеціальні методи, які здійснюють цю модифікацію. Назви таких методів прийнято починати зі слова set. На жаргоні такі методи інколи називають сетерами. Для читання властивостей, що оголошені в секції private створюються спеціальні методи, які безпечно повертають значення цих властивостей. Назви таких методів прийнято починати зі слова get. На жаргоні такі методи інколи називають ґетерами.

Члени класу, оголошені після ключового слова protected, стають захищеними членами. Вони доступні:

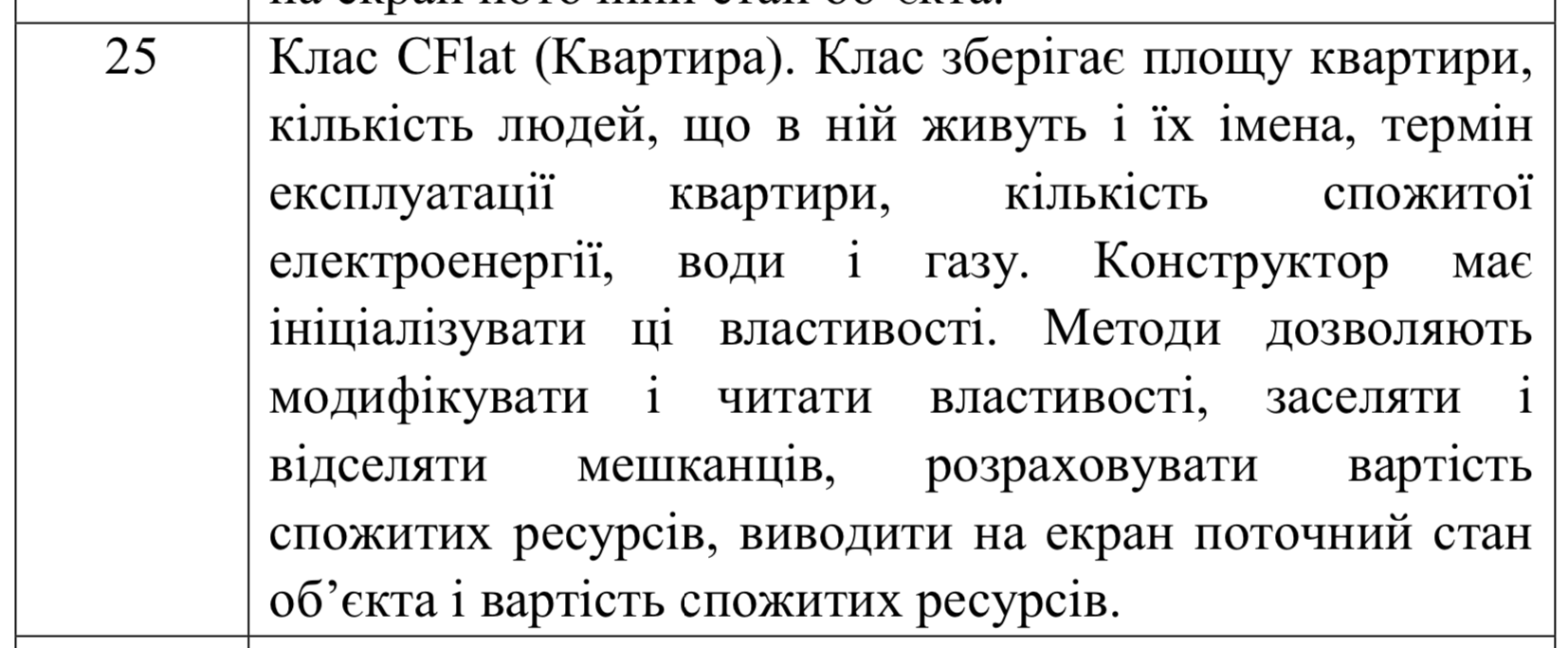
• для інших членів того ж класу; • членам похідних класів;  
• друзям класу.

При створенні об'єкта класу автоматично викликається спеціальний метод, що зветься конструктор. Конструктор – це метод класу ім‘я якого співпадає з іменем класу та не повертає ніяких значень (навіть void). Основне призначення конструктора – початкова ініціалізація об‘єкту класу.

У класі може бути визначено кілька перевантажених конструкторів, що відрізняються списком параметрів. При створенні об'єкта викликатися буде тільки один з них. Який саме – визначається зі способу створення об'єкта. Якщо конструктор має один параметр, то він може бути використаним для операції приведення типів. Якщо треба унеможливити використання конструктора для операції приведення типів, то його слід оголосити з використанням ключового слова explicit.

Конструктор без параметрів або конструктор, у якого всі аргументи – це аргументи за замовчуванням, має спеціальну назву – конструктор за замовчуванням. Конструктор за замовчуванням може бути лише один. Якщо в класі явно не визначений конструктор за замовчуванням, то він створюється автоматично і при виклику ініціалізуватиме всі члени-дані нулями. Приклади конструкторів за замовчуванням (клас може містити лише один з них)

**Індивідуальне завдання:**

****

**Код програми:**

#include <iostream>

#include <iomanip>

#include <vector>

#include <string>

#include <clocale>

**using** **namespace** std;

**class** CFlat

{

**private**:

vector <string> HumansName;

**int** time,

Square,

electricity,

water,

gas;

**public**:

CFlat(**int** set\_time, **int** set\_Square, **int** set\_electricity, **int** set\_water, **int** set\_gas)

{

setstate(set\_time, set\_Square, set\_electricity, set\_water, set\_gas);

}

**void** addHuman()

{

cout << "Введіть ім'я" << endl;

string name;

cin >> name;

HumansName.push\_back(name);

}

**void** viewHuman()

{

**for** (**unsigned** **int** i = 0; i < HumansName.size(); i++)

{

cout << i << "\t" << HumansName[i] << endl;

}

}

**int** getCountPeople()

{

**return** HumansName.size();

}

**void** delHuman()

{

viewHuman();

cout << "Введіть номер" << endl;

**int** k;

cin >> k;

HumansName.erase(HumansName.begin() + k);

}

**void** setstate(**int** time, **int** Square, **int** electricity, **int** water, **int** gas)

{

**this**->time = time;

**this**->Square = Square;

**this**->electricity = electricity;

**this**->water = water;

**this**->gas = gas;

}

**void** currentstate()

{

cout << "Термін експлуатації квартири: " << time << " дн" << endl;

cout << "Площа: " << Square << "м3" << endl;

cout << "Кількість спожитої електроенергії: " << electricity << " Вт/г" << endl;

cout << "Кількість спожитої води: " << water << " л" << endl;

cout << "Кількість спожитого газу: " << gas << " кб" << endl;

system("pause");

}

**double** price()

{

**return** electricity \* 1.2 + water \* 2.1 + gas \* 1.5;

}

};

CFlat N1(50, 15, 7, 50, 80);

**void** mainScreen()

{

system("cls");

cout << "Виберіть дію" << endl;

cout << "1. Заселити мешканця" << endl;

cout << "2. Переглянути імена мешканців" << endl;

cout << "3. Переглянути кількість мешканців" << endl;

cout << "4. Виселити мешканця" << endl;

cout << "5. Переглянути поточний стан" << endl;

cout << "6. Змінити стан" << endl;

cout << "7. Розрахувати вартість" << endl;

cout << "8. Вихід" << endl;

**int** ch;

cin >> ch;

**switch** (ch)

{

**case** 1:

system("cls");

N1.addHuman();

mainScreen();

**case** 2:

system("cls");

N1.viewHuman();

system("pause");

mainScreen();

**case** 3:

system("cls");

cout << "Кількість мешканців: " << N1.getCountPeople() << endl;

system("pause");

mainScreen();

**case** 4:

system("cls");

N1.delHuman();

mainScreen();

**case** 5:

system("cls");

N1.currentstate();

mainScreen();

**case** 6:

system("cls");

**int** time, Square, electricity, water, gas;

cout << "Введіть термін експлуатації квартири: " << endl;

cin >> time;

cout << "Введіть площу: " << endl;

cin >> Square;

cout << "Введіть кількість спожитої електроенергії: " << endl;

cin >> electricity;

cout << "Введіть кількість спожитої води: " << endl;

cin >> water;

cout << "Введіть кількість спожитого газу: " << endl;

cin >> gas;

N1.setstate(time, Square, electricity, water, gas);

mainScreen();

**case** 7:

system("cls");

cout << "Вартість спожитих ресурсів: " << N1.price() << "грн" << endl;

system("pause");

mainScreen();

**default**:

exit(0);

}

}

**int** main()

{

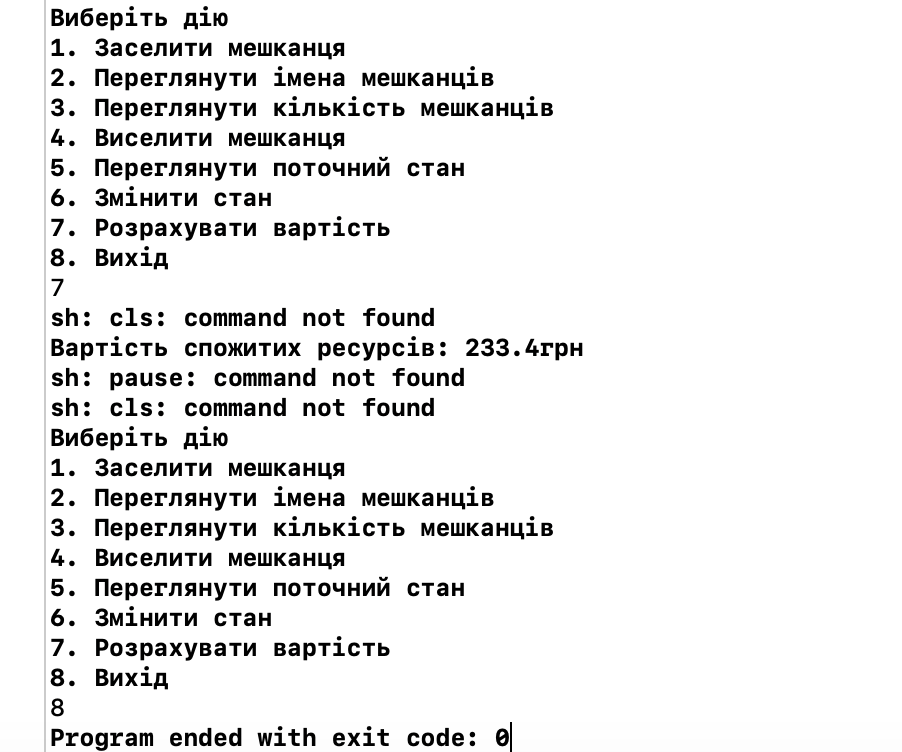
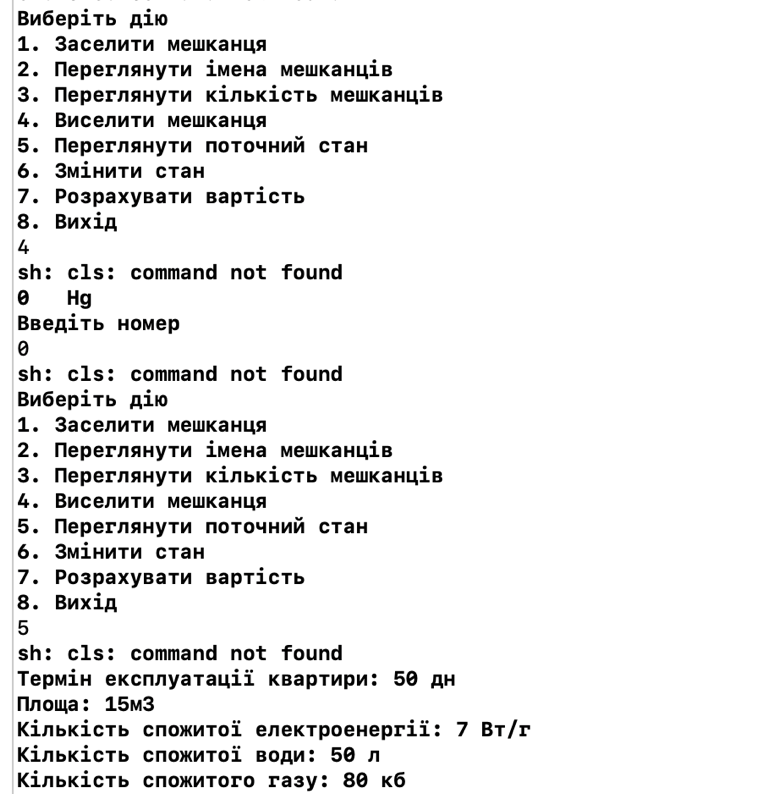
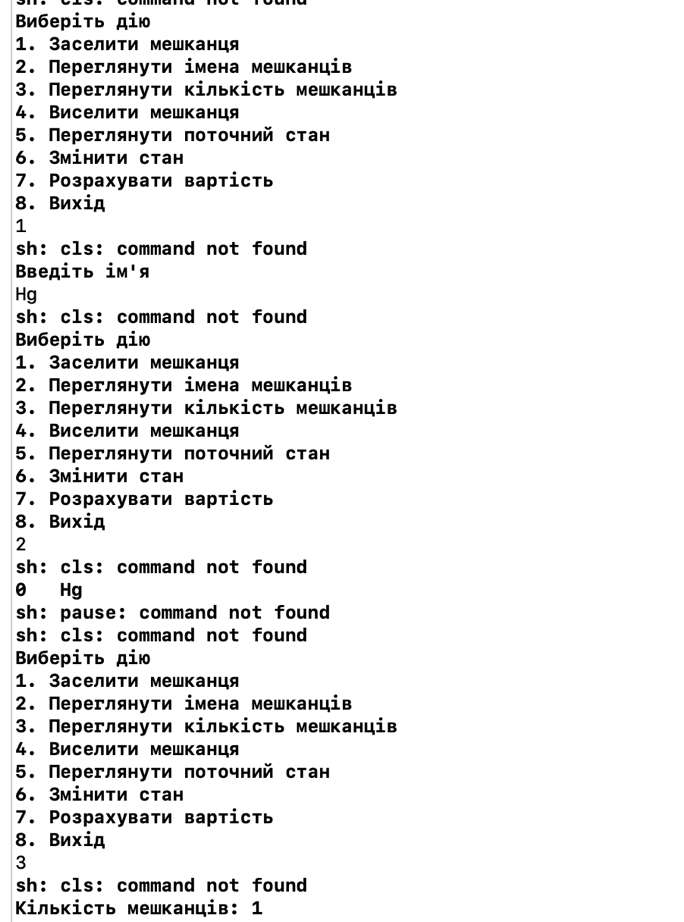
setlocale(LC\_ALL, "Russian");

mainScreen();

**return** 0;

}

**Результат виконання функції :**

****

**Висновок:** під час виконання цієї лабораторної роботи я навчилась працювати з класами та об’єктами.