

**Концепція об'єктно-орієнтованого програмування:
базові поняття та основні принципи**

Ознаки об'єктно-орієнтованої програми:

- ✓ в якості базових елементів використовуються класи, які породжують об'єкти;
- ✓ у процесі виконання програми може одночасно використовуватися кілька об'єктів, породжених від одного класу;
- ✓ класи організовано ієрархічно (ієрархія означає "бути частиною").

Дані та операції об'єкту:

- ✓ поля об'єкта (або атрибути вихідних даних), значення яких визначають поточний стан об'єкта;
- ✓ методи об'єкта, які реалізують дії (виконання алгоритмів) у відповідь на їх виклик у вигляді переданого повідомлення;
- ✓ властивості - частини методів, які визначають поведінку об'єкта, тобто його реакцію на зовнішні впливи.

Основні принципи ООП

АБСТРАГУВАНН
Я

ІНКАПСУЛЯЦІ
Я

СПАДКУВАННЯ

ПОЛІМОРФІЗМ

ОПИСАННЯ ОБ'ЄКТУ “БУДІВЛЯ”

Будівля

Поля:

Адреса

Площа

Власник

Рік_побудови

Вартість

Методи:

Вивести_на_екран()

Змінити_адресу()

Продати ()

Купити ()

Встановити_вартість ()

Абстрагування

ОБ'ЄКТ

СТРУКТУРОВАНА ЗМІННА, ЩО МІСТИТЬ ВСЮ ІНФОРМАЦІЮ ПРО ДЕЯКИЙ ФІЗИЧНИЙ ПРЕДМЕТ АБО ПОНЯТТЯ, ЩО РЕАЛІЗУЄТЬСЯ В ПРОГРАМІ

**Властивос
ті**

*Параметри об'
єкту*

Методи

*Процедури та функції,
що реалізуються об'
єктом*

Інкапсуляція

ОПИСАННЯ КЛАСУ “БУДІВЛЯ”

```
class Building {  
  
  private:  
  
  String address;  
  
  float area;  
  
  String owner;  
  
  int cons_year;  
  
  float coast;  
  
  public:  
  
  void show();  
  
  void change_address (String);  
  
  void to_sell (String ow, float cos);  
  
  void to_buy (String ow, float cos);  
  
  float SetCoast (float );    }  
}
```

```
class Building {
```

```
private:
```

```
String address;
```

```
float area;
```

```
String owner;
```

```
int cons_year;
```

```
float coast;
```

```
public:
```

```
void show();
```

```
void change_address (String);
```

```
void to_sell (String ow, float  
cos);
```

```
void to_buy (String ow, float  
cos);
```

```
float SetCoast (float ); } }
```

```
class House: Building {
```

```
private:
```

```
int count_floors;
```

```
int count_apart;
```

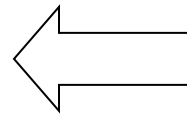
```
int count_lodgers;
```

```
public:
```

```
void show();
```

```
void change_lodgers (int);
```

```
}
```

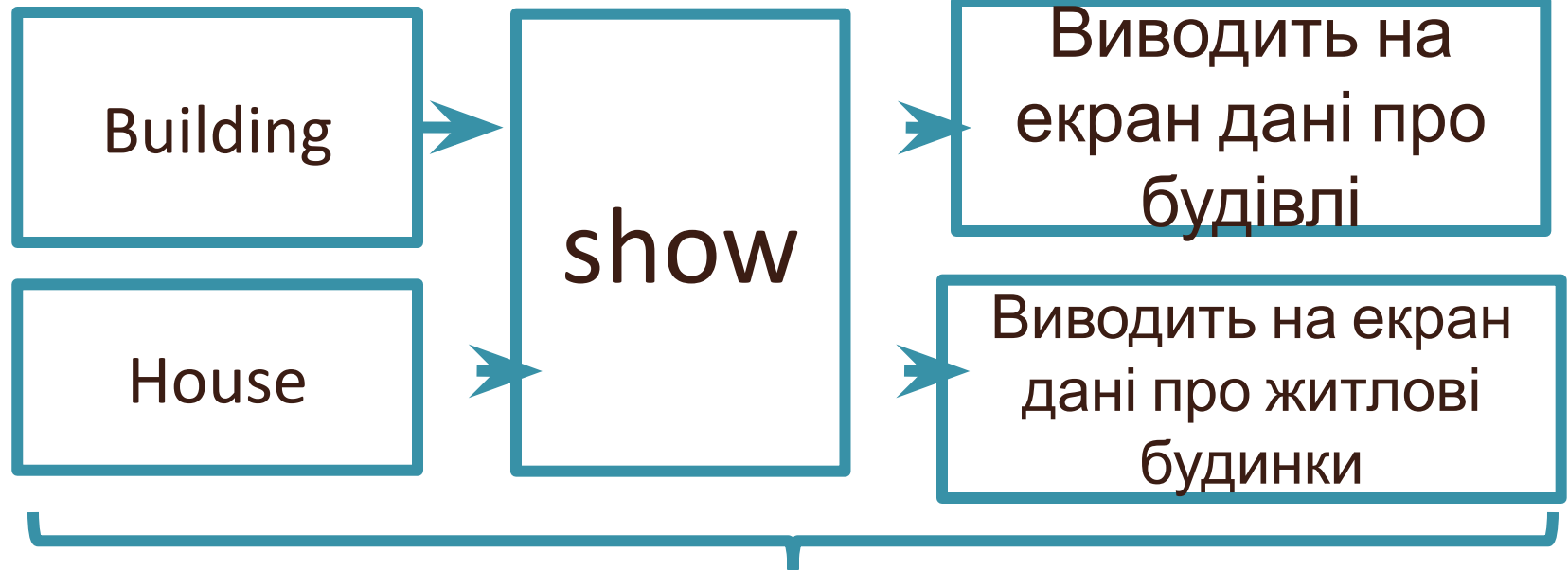


Спадкування

Класи

Метод

Дія



Поліморфізм