

**МІНІСТЕРСТВО**  
**ОСВІТИ ТА НАУКИ УКРАЇНИ**  
**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Кафедра інформаційних систем та технологій**

**Звіт**  
**з лабораторної роботи № 4**  
**«Інтерфейси. Абстрактні класи»**  
**з дисципліни**  
**«Програмування – 2. ООП»**

**Варіант №21**

**Перевірив:**

**доц. Корнага Ярослав**  
**Ігорович**

**Виконала:**

**Студентка ІС-13, ФІОТ**  
**Росновська Ольга**

**Київ 2022**

## Завдання

- Проаналізувати завдання, виділити інформаційні об'єкти та дії. Визначити правильну ієрархію об'єктів
- Створити базовий суперклас (абстрактний клас або інтерфейс) і визначити загальні методи для даного класу. Створити підкласи, в які додати специфічні властивості та методи. Частина методів перевизначити.
- Розробити програму з використанням абстрактних класів та інтерфейсів. Чітко розуміти, де доцільно використати суперклас, а де звичайний.
- При розробці використовувати наслідування та поліморфізм
- У всіх класах повинні бути реалізовані доцільні для класу методи, навіть якщо це не вказано у завданні
- Використовувати об'єкти підкласів для моделювання реальних ситуацій на об'єктів

Створити суперклас Будівля і підкласи Громадська Будівля, Житлова Будівля, Кінотеатр, Готель. За допомогою конструктора задати дату побудови будівлі. Визначити вартість оренди одного місця в будівлі. Визначити вартість вхідного квитка в громадську будівлю. Визначити місткість будівлі.

## Program.cs

```
using System;
using System.Collections.Generic;

namespace Lab_4
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.OutputEncoding = System.Text.Encoding.Unicode;
            Cinema florencia = new Cinema("23.08.2001", 125, 4);
            Console.WriteLine(florencia);
            Console.WriteLine("Дата побудови: " + florencia.Date);
            Console.WriteLine("Вартість квитка (пільговий): " +
florencia.CountPrice(50, 3));
            Console.WriteLine("Вартість квитка (звичайний): " +
florencia.CountPrice(50, 30));
            Console.WriteLine("Місткість: " + florencia.capacity());

            Hotel California = new Hotel("10.02.1987", 500, 8);
            Console.WriteLine("\n" + California);
            Console.WriteLine("Дата побудови: " + California.Date);
            Console.WriteLine("Вартість оренди: " + California.CountRent(125, 4,
12));
            Console.WriteLine("Місткість: " + California.capacity());

            PublicBuilding museum = new PublicBuilding("05.12.1962", 180, 3);
```

```

        Console.WriteLine("\n" + museum);
        Console.WriteLine("Дата побудови: " + museum.Date);
        Console.WriteLine("Вартість квитка (звичайний): " +
museum.CountPrice(100, 34));
        Console.WriteLine("Вартість квитка (пільговий): " +
museum.CountPrice(100, 84));
        Console.WriteLine("Місткість: " + museum.capacity());

        ResidentialBuilding house = new ResidentialBuilding("26.01.2015", 175,
19);

        Console.WriteLine("\n" + house);
        Console.WriteLine("Дата побудови: " + house.Date);
        Console.WriteLine("Вартість оренди: " + house.CountRent(175, 45, 28));
        Console.WriteLine("Місткість: " + house.capacity());

        List<Building> building_list = new List<Building> { florencia,
California,
museum };
        City misto = new City(building_list);
        misto.show_building();
        misto.add_building(house);
        misto.show_building();
        misto.average_capacity();
    }
}
}

```

## buildings.cs

```
using System;
using System.Collections.Generic;

public interface IRent
{
    public int CountRent(int price, int quantity, int days);
}

public interface ITicket
{
    public int CountPrice(int price, int customer_age);
}

public abstract class Building
{
    public string Date { get; set; }
    public int Square { get; set; }
    public int Floors { get; set; }
    public Building(string date, int square, int floors)
    {
        this.Date = date;
        this.Square = square;
        this.Floors = floors;
    }

    public abstract int capacity();
}

public class PublicBuilding : Building, ITicket
{
    public PublicBuilding(string date, int square, int floors) : base(date,
square, floors) { }

    public int CountPrice(int price, int customer_age)
    {
        if (customer_age <= 6 || customer_age >= 60)
        {
            return price / 2;
        }
        return price;
    }

    public override int capacity()
    {
        return (this.Square * this.Floors) / 5;
    }
}

public class ResidentialBuilding : Building, IRent
{
    public ResidentialBuilding(string date, int square, int floors) : base(date,
square, floors) { }

    public int CountRent(int price_per_meter, int square, int days)
    {
        return price_per_meter * square * days;
    }

    public override int capacity()
    {
        return (this.Square * this.Floors) / 8;
    }
}
```

```

    }
}

public class Cinema : PublicBuilding, ITicket
{
    public Cinema(string date, int square, int floors) : base(date, square,
floors) { }
    public int Price { get; set; }

    public override int capacity()
    {
        return (this.Square * this.Floors) / 4;
    }
}

public class Hotel : ResidentialBuilding, IRent
{
    public Hotel(string date, int square, int floors) : base(date, square,
floors) { }

    public override int capacity()
    {
        return (this.Square * this.Floors) / 7;
    }
}

public class City
{
    public List<Building> Buildings;
    public City(List<Building> building_list)
    {
        this.Buildings = building_list;
    }

    public void add_building(Building building)
    {
        Buildings.Add(building);
    }

    public void show_building()
    {
        foreach (Building b in Buildings)
        {
            Console.WriteLine("\n" + b);
        }
    }

    public void average_capacity()
    {
        int cinema_sum = 0;
        int cinema_count = 0;
        int hotel_sum = 0;
        int hotel_count = 0;
        int public_sum = 0;
        int public_count = 0;
        int residential_sum = 0;
        int residential_count = 0;
        foreach (Building b in Buildings)
        {
            if (b is Cinema)
            {
                cinema_sum += b.capacity();
                cinema_count += 1;
            }
        }
    }
}

```

```

        if (b is Hotel)
        {
            hotel_sum += b.capacity();
            hotel_count += 1;
        }

        if (b is PublicBuilding)
        {
            public_sum += b.capacity();
            public_count += 1;
        }

        if (b is ResidentialBuilding)
        {
            residential_sum += b.capacity();
            residential_count += 1;
        }
    }
    if (cinema_count != 0)
    {
        Console.WriteLine("\n" + "Середня ємність кінотеатрів: " +
cinema_sum / cinema_count);
    }
    if (public_count != 0)
    {
        Console.WriteLine("\n" + "Середня ємність громадських будівель:
" + public_sum / public_count);
    }
    if (residential_count != 0)
    {
        Console.WriteLine("\n" + "Середня ємність всіх житлових
будівель: " + residential_sum / residential_count);
    }
    if (hotel_count != 0)
    {
        Console.WriteLine("\n" + "Середня ємність всіх готелів: " +
hotel_sum / hotel_count);
    }
}
}

```

## Результат виконання програми

Cinema

Дата побудови: 23.08.2001

Вартість квитка (пільговий): 25

Вартість квитка (звичайний): 50

Місткість: 125

Hotel

Дата побудови: 10.02.1987

Вартість оренди: 6000

Місткість: 571

PublicBuilding

Дата побудови: 05.12.1962

Вартість квитка (звичайний): 100

Вартість квитка (пільговий): 50

Місткість: 108

ResidentialBuilding

Дата побудови: 26.01.2015

Вартість оренди: 220500

Місткість: 415

Cinema

Hotel

PublicBuilding

Cinema

Hotel

PublicBuilding

ResidentialBuilding

Середня ємність кінотеатрів: 125

Середня ємність громадських будівель: 116

Середня ємність всіх житлових будівель: 493

Середня ємність всіх готелів: 571

## **Висновок**

Під час виконання цієї лабораторної роботи ми ознайомились з абстрактними класами.