

Введение в классы

Листинг 1

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

// -----
// ОПРЕДЕЛЕНИЕ КЛАССА
// -----

namespace ConsoleApp_Classes1
{
    // Общая форма определения класса Building
    // Определение класс создает новый тип данных
    class Building
    {
        // Объявление переменных класса Building
        // доступ тип переменная1;
        public int floors;           // количество этажей
        public int area;             // общая площадь основания здания
        public int occupants;        // количество жильцов
    }

    class Program
    {
        static void Main(string[] args)
        {
            Building house1 = new Building();    // создается объект
            double areaPerOccupants;             // площадь,
            // Приписываем значения полям объекта house1
            house1.floors = 4;
            house1.area = 2500;
            house1.occupants = 4;

            // Вычисляем площадь, приходящуюся на одного жильца дома
            areaPerOccupants = house1.area / house1.occupants;

            Console.WriteLine("Дом общей площадью {0} имеет {1} этажа и
{2} жильцов", house1.area, house1.floors, house1.occupants);
            Console.WriteLine("На одного человека приходится {0} кв.
футов", areaPerOccupants);
            Console.ReadLine();
        }
    }
}
```

Листинг 2

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

// -----
// МЕТОДЫ КЛАССА
// -----

namespace ConsoleApp_Classes1
{
    // Общая форма определения класса Building
    // Определение класс создает новый тип данных
    class Building
    {
        // Объявление переменных класса Building
        // доступ тип переменная1;
        public int floors;        // количество этажей
        public int area;          // общая площадь основания здания
        public int occupants;     // количество жильцов

        // Объявление методов Building
        // доступ тип_возврата метод1 (параметры);
        // {
        // тело метода
        // }
        // Вычисляем площадь, приходящуюся на одного человека
        public int areaPerOccupants()
        {
            // Вычисляем и возвращаем площадь, приходящуюся на одного
            // жильца
            return area / occupants;
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            Building house = new Building();    // создается объект
            Building office = new Building();
            int areaPO;                          // площадь, приходящаяся на одного
            // жильца

            // Присваиваем значения полям объекта house
            house.floors = 4;
            house.area = 2500;
            house.occupants = 4;
        }
    }
}
```

```
// Присваиваем значения полям объекта office
office.floors = 3;
office.area = 4200;
office.occupants = 25;

Console.WriteLine("Дом общей площадью {0} имеет {1} этажа и
{2} жильцов", house.area, house.floors, house.occupants);
areaPO = house.areaPerOccupants();
Console.WriteLine("На одного человека приходится {0} кв.
футов", areaPO);
Console.ReadLine();

Console.WriteLine("Офис общей площадью {0} имеет {1} этажа и
{2} работника", office.area, office.floors, office.occupants);
areaPO = office.areaPerOccupants();
Console.WriteLine("На одного человека приходится {0} кв.
футов", areaPO);
Console.ReadLine();
    }
}
```

Листинг 3

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

// -----
// ПАРАМЕТРИЗОВАННЫЕ МЕТОДЫ КЛАССА
// -----

namespace ConsoleApp_Classes1
{
    // Общая форма определения класса Building
    // Определение класса создает новый тип данных
    class Building
    {
        // Объявление переменных класса Building
        public int floors;      // количество этажей
        public int area;        // общая площадь основания здания
        public int occupants;    // количество жильцов

        // Вычисляем и возвращаем площадь, приходящуюся на одного жильца
        public int areaPerOccupants()
        {
            return area / occupants;
        }

        // Вычисляем и возвращаем максимальное количество человек,
        // если на каждого человека должна приходиться заданная площадь
        public int maxOccupants(int minArea)
        {
            return area / minArea;
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            Building house = new Building();    // создается объект Building
            Building office = new Building();
            double areaPO;                      // площадь, приходящаяся на одного обитателя

            // Присваиваем значения полям объекта house
            house.floors = 4;
            house.area = 2500;
            house.occupants = 4;

            // Присваиваем значения полям объекта office
            office.floors = 3;
            office.area = 4200;
            office.occupants = 25;

            Console.WriteLine("Дом общей площадью {0} имеет {1} этажа и {2}
жильцов", house.area, house.floors, house.occupants);
            areaPO = house.areaPerOccupants();
            //Console.WriteLine("На одного человека приходится {0} кв. футов",
areaPO);

            Console.WriteLine("Максимальное кол-во человек {0}, если на одного
человека должно приходится {1} кв. футов", house.maxOccupants(300), "300");
            Console.ReadLine();
        }
    }
}
```

```
        Console.WriteLine("Офис общей площадью {0} имеет {1} этажа и {2}  
работника", office.area, office.floors, office.occupants);  
        areaPO = office.areaPerOccupants();  
        //Console.WriteLine("На одного человека приходится {0} кв. футов",  
areaPO);  
        Console.WriteLine("Максимальное кол-во человек {0}, если на одного  
человека должно приходится {1} кв. футов", office.maxOccupants(100), "100");  
        Console.ReadLine();  
    }  
}
```

Листинг 4

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

// -----
// КОНСТРУКТОР КЛАССА (параметризированный)
// -----

namespace ConsoleApp_Classes4
{
    // Общая форма определения класса Building
    // Определение класс создает новый тип данных
    class Building
    {
        // Объявление переменных класса Building
        public int floors;        // количество этажей
        public int area;          // общая площадь основания здания
        public int occupants;     // количество жильцов

        // Добавление конструктора класса Building
        public Building(int f, int a, int o)
        {
            floors = f;
            area = a;
            occupants = o;
        }

        // Вычисляем и возвращаем площадь, приходящуюся на одного жильца
        public int areaPerOccupants()
        {
            return area / occupants;
        }

        // Вычисляем и возвращаем максимальное количество человек,
        // если на каждого человека должна приходиться заданная площадь
        public int maxOccupants(int minArea)
        {
            return area / minArea;
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            // Используем параметризированный конструктор класса Building
            Building house = new Building(2, 2500, 4);
            Building office = new Building(3, 4200, 25);
            int areaPO;          // площадь, приходящаяся на одного
            // жильца
        }
    }
}
```

```
Console.WriteLine("Дом общей площадью {0} имеет {1} этажа и  
{2} жильцов", house.area, house.floors, house.occupants);  
areaPO = house.areaPerOccupants();  
//Console.WriteLine("На одного человека приходится {0} кв.  
футов", areaPO);  
Console.WriteLine("Максимальное кол-во человек {0}, если на  
одного человека должно приходится {1} кв. футов", house.maxOccupants(300),  
"300");  
Console.ReadLine();  
  
Console.WriteLine("Офис общей площадью {0} имеет {1} этажа и  
{2} работника", office.area, office.floors, office.occupants);  
areaPO = office.areaPerOccupants();  
//Console.WriteLine("На одного человека приходится {0} кв.  
футов", areaPO);  
Console.WriteLine("Максимальное кол-во человек {0}, если на  
одного человека должно приходится {1} кв. футов",  
office.maxOccupants(200), "200");  
Console.ReadLine();  
    }  
}  
}
```

