

C++

Урок 13

A hand-drawn blue oval frame with a double-line border, centered on the page. The text is written in the center of this frame.

Разбор ДЗ

Разбор дз

1 задача:

Задача: Доделайте задачу из работы на занятии.
Написать функцию `resize()`, которая изменяет размер динамического массива.

Разбор дз

1 задача:

```
int main()
{

    int size, newS;
    cin >> size;

    int *ptr = read(size);
    show(ptr, size);
    cout << endl;
    cin >> newS;

    ptr = resize(ptr, size, newS);
    show(ptr, size);

    delete[] ptr;
}
```

main()

Разбор дз

1 задача:

```
int *resize(int *arr, int &size, const int &newSize)
{
    size = newSize;

    int *ptr = new int[size];
    for (int i = 0; i < size; i++)
    {
        ptr[i] = arr[i];
    }
    delete[] arr;
    return ptr;
}
```

Реализация функции (вар 1)

Разбор дз

1 задача:

```
int *resize(int *arr, int &size, const int &newSize)
{
    if (size >= newSize)
    {
        size = newSize;

        int *ptr = new int[size];
        for (int i = 0; i < size; i++)
        {
            ptr[i] = arr[i];
        }
        delete[] arr;
        return ptr;
    }
    else
    {
        // size = newSize;
        int *ptr = new int[newSize];
        for (int i = 0; i < size; i++)
        {
            ptr[i] = arr[i];
        }
        for (int i = size; i < newSize; i++)
        {
            ptr[i] = 0;
        }
        size = newSize;
        delete[] arr;
        return ptr;
    }
}
```

Реализация функции (вар 2)

Разбор дз

2 задача:

Задача: Напишите функцию `split()`, которая разбивает число на разряды и заносит его в динамический массив.

Разбор дз

2 задача:

```
int main()
{

    int num;
    cin >> num;

    int col = count(num); //функция подсчета количесива разрядов
    int *ptr = split(num); //функция записи в массив
    show(ptr, col);        //функция вывода

    delete[] ptr;

}
```

main() функция

Разбор дз

2 задача:

```
int count(const int &num)
{
    int temp = num;
    int col = 0;
    while (temp)
    {
        col++;
        temp /= 10;
    }
    return col;
}

int *split(const int &num)
{
    int buf = num;
    int size = count(num);
    int *ptr = new int[size];
    for (int i = 0; i < size; i++)
    {
        ptr[size - i - 1] = buf % 10;
        buf /= 10;
    }
    return ptr;
}
```

Реализация функций

Разбор дз

3 - 4 задача:

Задача: Напишите функцию `find()/rfind()`, которая возвращает индекс переданной переменной в одномерном динамическом массиве. (Если таких элементов больше одного, то вывести индекс **первого / последнего** встречаемого)

Разбор дз

3 – 4 задача:

```
int rfind(int *A, const int &num, const int &size)
{
    int index = 0;
    for (int i = 0; i < size; i++)
        if (A[i] == num)
            index = i;

    return index;
}

int find(int *A, const int &num, const int &size)
{
    int index = 0;
    for (int i = size - 1; i >= 0; i--)
        if (A[i] == num)
            index = i;

    return index;
}
```

Реализация функций

Разбор дз

5 задача:

Задача: Напишите функцию `unique()`, которая удаляет повторяющиеся подряд элементы в одномерном динамическом массиве.

Разбор дз

5 задача:

```
int main()
{

    int size, val;
    cin >> size;

    int *ptr = read(size);
    show(ptr, size);
    cout << endl;
    ptr = unique(ptr, size);
    show(ptr, size);

}
```

main() функция

Разбор дз

5 задача:

```
int *unique(int *A, int &size)
{
    int cnt = 0;

    for (int i = 0; i < size; i++)
    {
        if (A[i] == A[i + 1])
        {
            A[i] = 0;
            cnt++;
        }
    }

    int *ptr = new int[cnt];
    int temp = 0;
    for (int i = 0; i < size; i++)
    {
        if (A[i] != 0)
        {
            ptr[temp] = A[i];
            temp++;
        }
    }
    size = temp;
    delete[] A;
    return ptr;
}
```

Функция unique()

A hand-drawn blue oval frame with a slightly irregular, sketchy border, centered on the page. The word "Повторение" is written inside this frame in a bold, black, sans-serif font.

Повторение

**Что такое динамический
массив?**

Где он хранится?

Где хранится указатель на него?

**Имя массива это указатель на
какой элемент?**

**В чем отличие указателей от
ссылок?**

**Расскажите алгоритм написания
функции, возвращающая
указатель на дин массив**



Объектное программирование

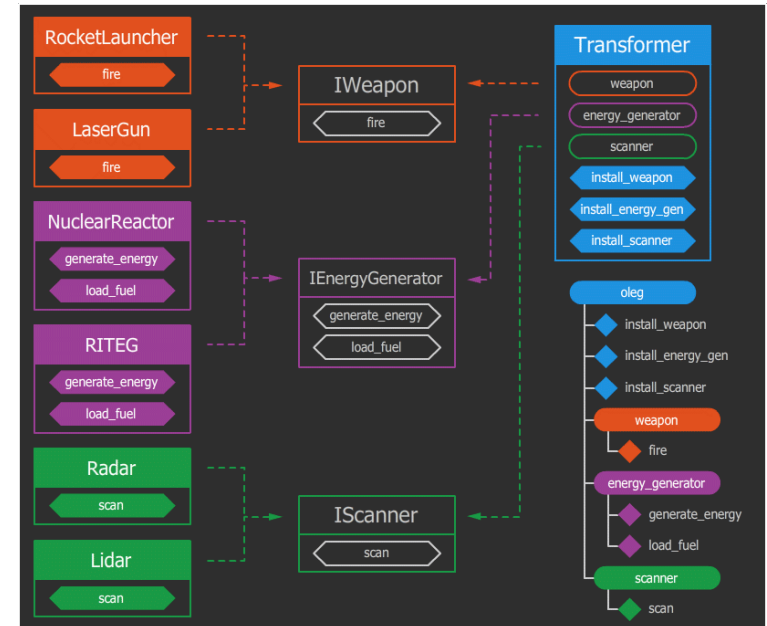
ООП

Переход от объемного кода к компактности

```
int count(const int &num)
{
    int temp = num;
    int col = 0;
    while (temp)
    {
        col++;
        temp /= 10;
    }
    return col;
}

int *split(const int &num)
{
    int buf = num;
    int size = count(num);
    int *ptr = new int[size];
    for (int i = 0; i < size; i++)
    {
        ptr[size - i - 1] = buf % 10;
        buf /= 10;
    }
    return ptr;
}
```

Функциональный стиль



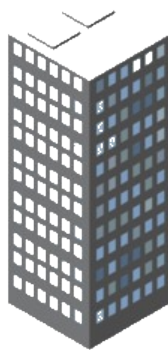
Объектный стиль

ООП

Представьте себе жилой комплекс.

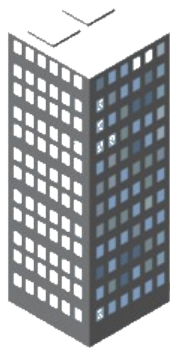
ООП

Представьте себе жилой комплекс.



ООП

Представьте себе жилой комплекс.



ООП

Представьте себе жилой комплекс.





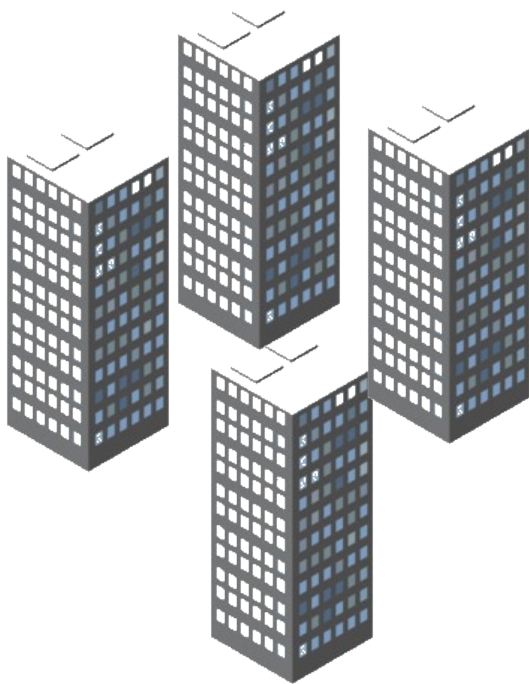
Как он был построен?

Похожи ли здания внутри ЖК?

Могут ли они отличаться?

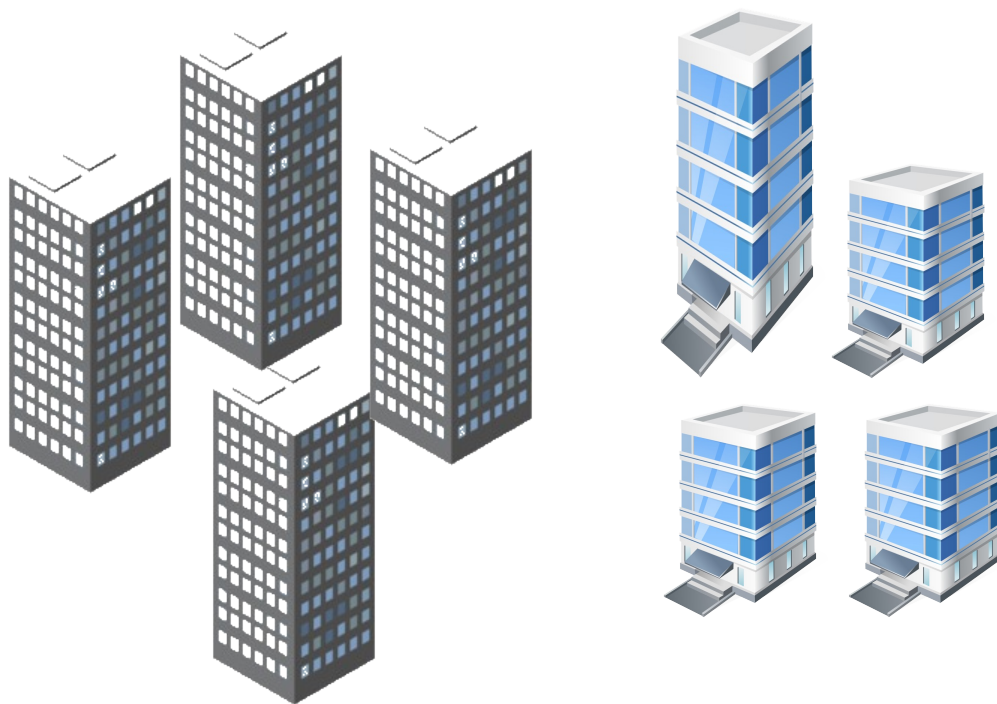
ООП

Заглянем в разные точки города



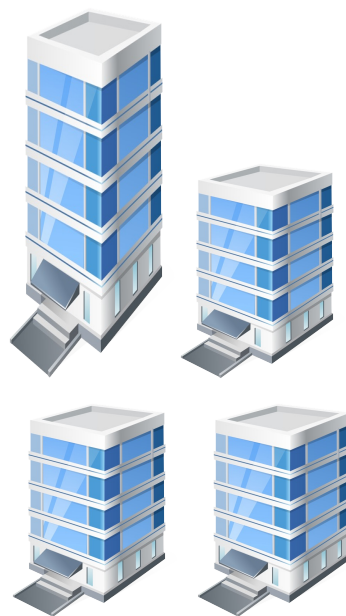
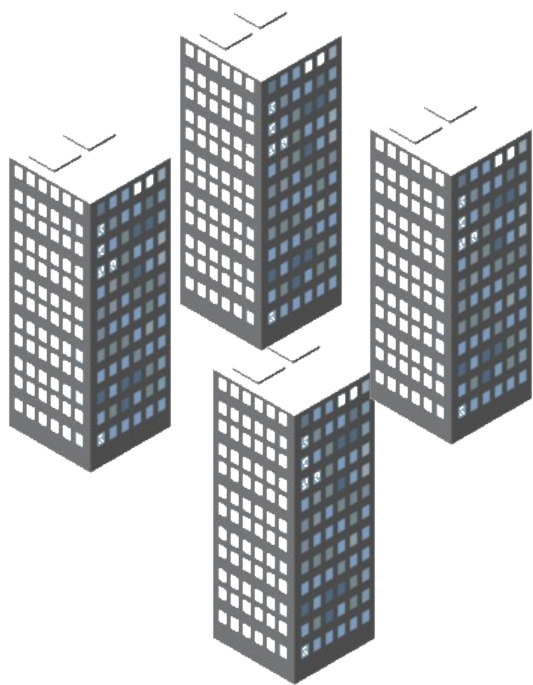
ООП

Заглянем в разные точки города



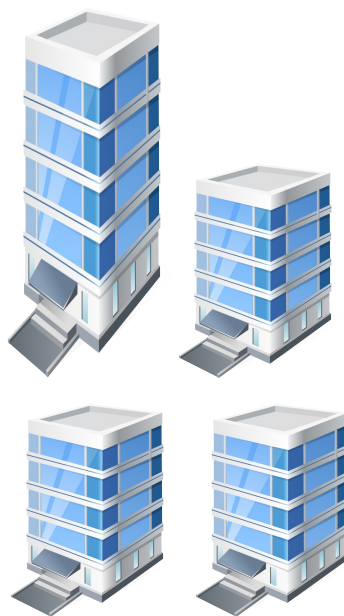
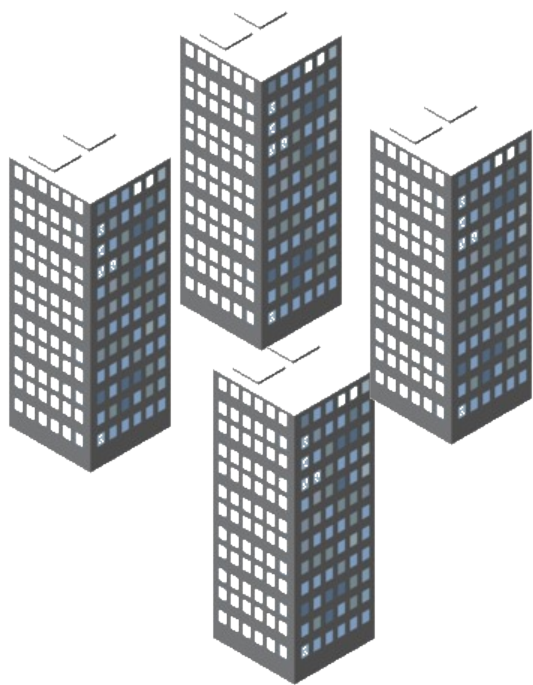
ООП

Заглянем в разные точки города



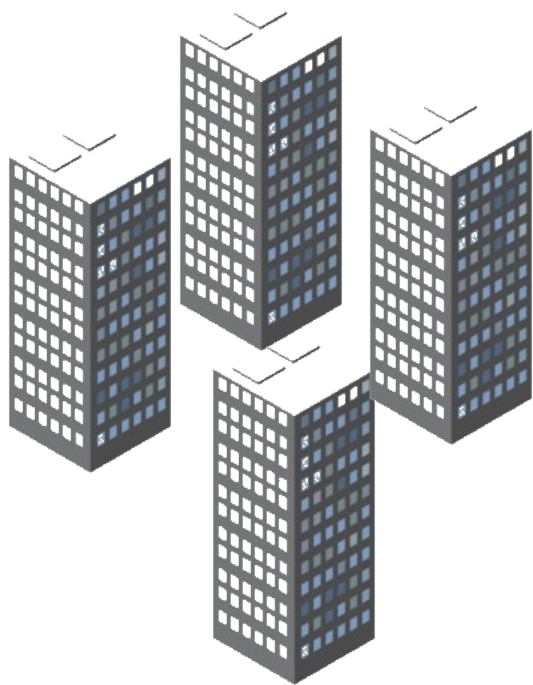
ООП

Заглянем в разные точки города

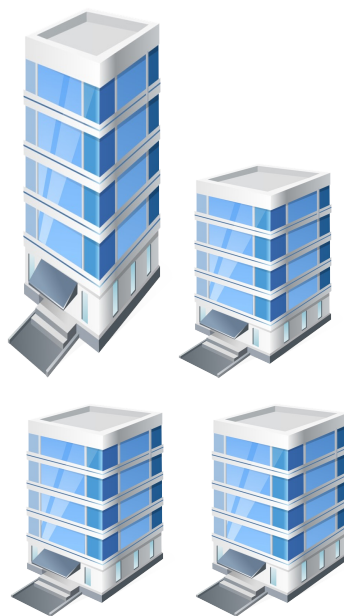


ЖК1

Заглянем в разные точки города



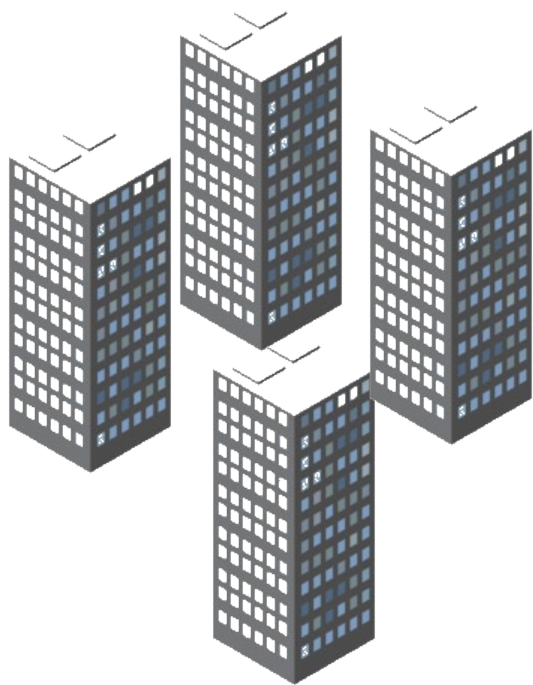
жк1



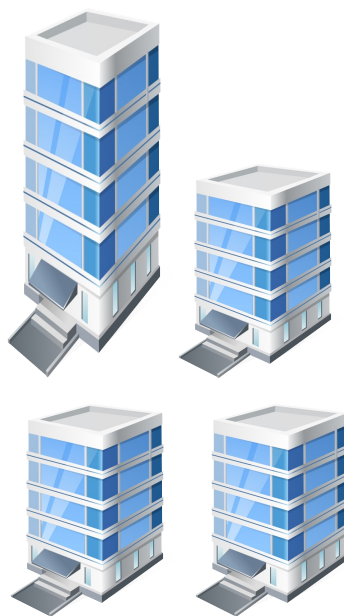
жк2



Заглянем в разные точки города



жк1



жк2

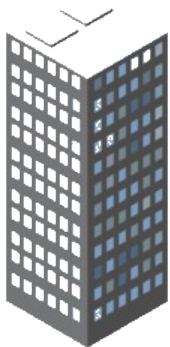


жк3

**Отличаются ли между собой
ЖК?**

ООП

Рассмотрим их особенности



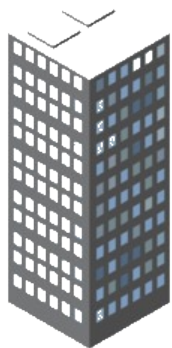
- Название;
- Кол-во этажей;
- Вертолетная площадка;
- Может подсвечиваться().

жк1

жк2

жк3

Рассмотрим их собенности



- Название;
- Кол-во этажей;
- Вертолетная площадка;
- Может подсвечиваться().

жк1



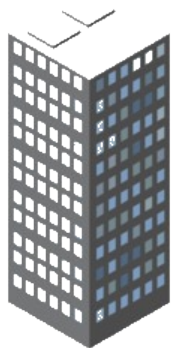
- Название;
- Кол-во этажей;
- Год постройки;

жк2



жк3

Рассмотрим их особенности



- Название;
- Кол-во этажей;
- Вертолетная площадка;
- Может подсвечиваться().

жк1



- Название;
- Кол-во этажей;
- Год постройки.

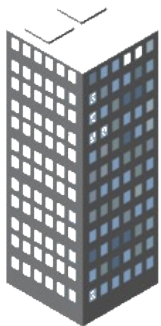
жк2



- Название;
- Кол-во этажей;
- Материал;
- Цвет;
- Сигнализация().

жк3

ООП



- Название;
- Кол-во этажей;
- Вертолетная площадка;
- Может подсвечиваться().

жк1



- Название;
- Кол-во этажей;
- Год постройки.

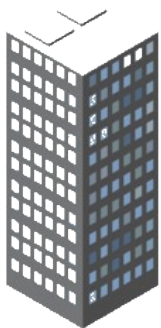
жк2



- Название;
- Кол-во этажей;
- Материал;
- Цвет;
- Сигнализация().

жк3

ООП



- Название;
- Кол-во этажей;
- Вертолетная площадка;
- Может подсвечиваться().

жк1



- Название;
- Кол-во этажей;
- Год постройки.

жк2

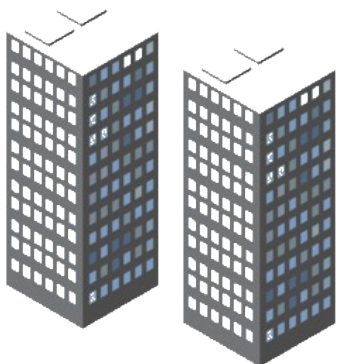


- Название;
- Кол-во этажей;
- Материал;
- Цвет;
- Сигнализация().

жк3

← классы →

ООП



- Название;
- Кол-во этажей;
- Вертолетная площадка;
- Может подсвечиваться().

жк1



- Название;
- Кол-во этажей;
- Год постройки.

жк2

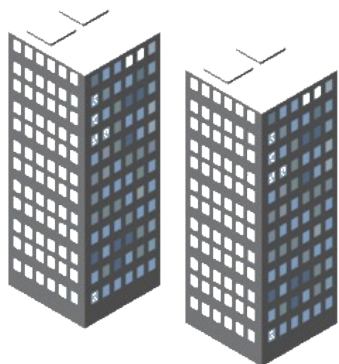


- Название;
- Кол-во этажей;
- Материал;
- Цвет;
- Сигнализация().

жк3

← классы →

ООП



- Название;
- Кол-во этажей;
- Вертолетная площадка;
- Может подсвечиваться().

жк1



- Название;
- Кол-во этажей;
- Год постройки.

жк2

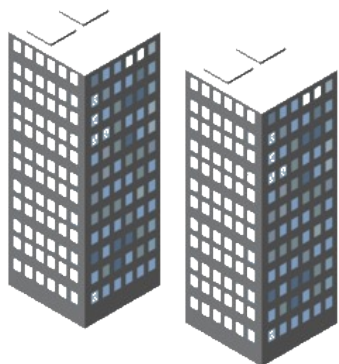


- Название;
- Кол-во этажей;
- Материал;
- Цвет;
- Сигнализация().

жк3

классы

ООП



- Название;
- Кол-во этажей;
- Вертолетная площадка;
- Может подсвечиваться().

жк1



- Название;
- Кол-во этажей;
- Год постройки.

жк2

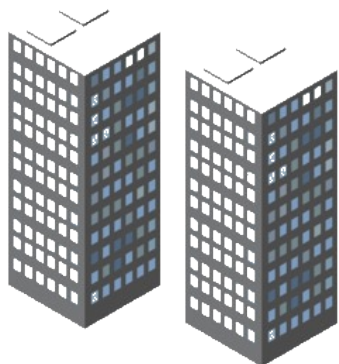


- Название;
- Кол-во этажей;
- Материал;
- Цвет;
- Сигнализация().

жк3

классы

ООП



- Название;
- Кол-во этажей;
- Вертолетная площадка;
- Может подсвечиваться().

жк1



- Название;
- Кол-во этажей;
- Год постройки.

жк2

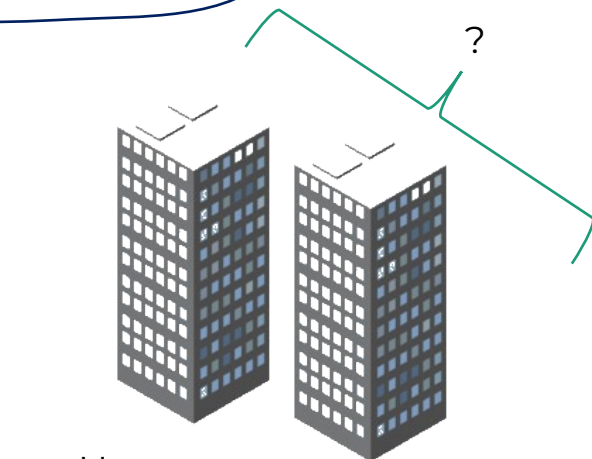


- Название;
- Кол-во этажей;
- Материал;
- Цвет;
- Сигнализация().

жк3

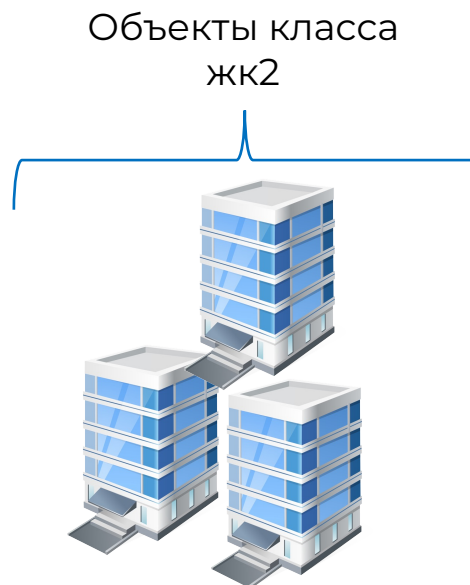
классы

ООП



- Название;
- Кол-во этажей;
- Вертолетная площадка;
- Может подсвечиваться().

жк1



- Название;
- Кол-во этажей;
- Год постройки.

жк2



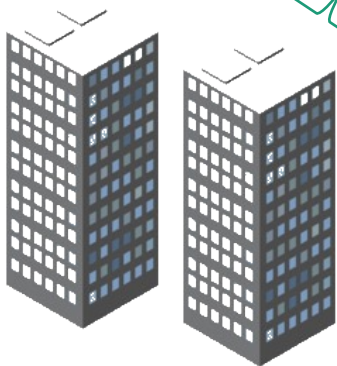
- Название;
- Кол-во этажей;
- Материал;
- Цвет;
- Сигнализация().

жк3

классы

ООП

Объект класса
жк1



- Название;
- Кол-во этажей;
- Вертолетная площадка;
- Может подсвечиваться().

жк1

Объекты класса
жк2



- Название;
- Кол-во этажей;
- Год постройки.

жк2

Объект класса
жк3



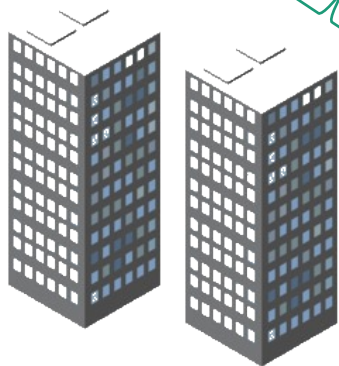
- Название;
- Кол-во этажей;
- Материал;
- Цвет;
- Сигнализация().

жк3

классы

ООП

Объект класса
жк1



- Название;
- Кол-во этажей;
- Вертолетная площадка;
- Может подсвечиваться().

жк1

Объект класса
жк3



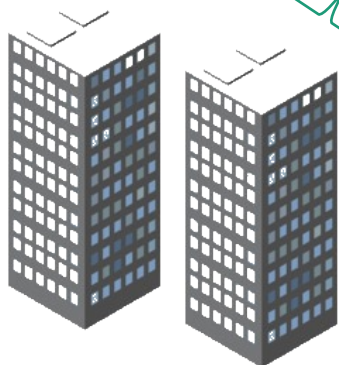
- Название;
- Кол-во этажей;
- Материал;
- Цвет;
- Сигнализация().

жк3

→ классы ←

ООП

Объект класса
жк1



- Название;
- Кол-во этажей;
- Вертолетная площадка;
- Может подсвечиваться().

поля класса
жк1

жк1

Объект класса
жк3



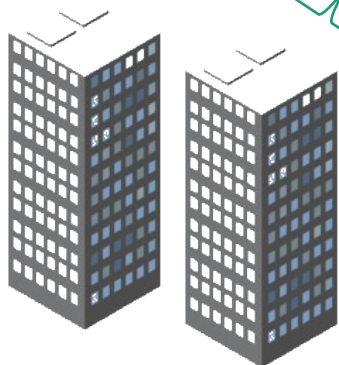
- Название;
- Кол-во этажей;
- Материал;
- Цвет;
- Сигнализация().

жк3

← классы →

ООП

Объект класса
жк1



- Название;
- Кол-во этажей;
- Вертолетная площадка;

поля класса
жк1

- Может
подсвечиваться().

методы класса
жк1

жк1

Объект класса
жк3



- Название;
- Кол-во этажей;
- Материал;
- Цвет;

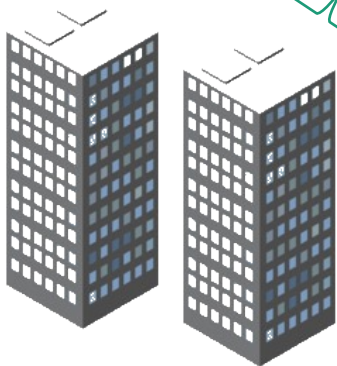
- Сигнализация().

жк3

← классы →

ООП

Объект класса
жк1



- Название;
- Кол-во этажей;
- Вертолетная площадка;

поля класса
жк1

- Может
подсвечиваться().

методы класса
жк1

жк1

Объект класса
жк3



?

- Название;
- Кол-во этажей;
- Материал;
- Цвет;

?

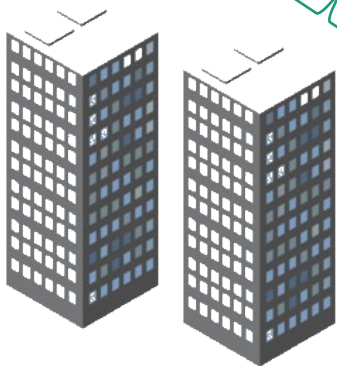
- Сигнализация().

жк3

← классы →

ООП

Объект класса
жк1



- Название;
- Кол-во этажей;
- Вертолетная площадка;

поля класса
жк1

- Может
подсвечиваться().

методы класса
жк1

жк1

Объект класса
жк3



- Название;
- Кол-во этажей;
- Материал;
- Цвет;

поля класса
жк3

- Сигнализация().

методы класса
жк3

жк3

← классы →

ООП

Объект — это сущность, способная сохранять свое состояние (информацию) и обеспечивающая набор операций (поведение) для проверки и изменения этого состояния.