



C++

Разбор домашней работы



Указатель this

Теория



15 минут



Что такое указатель?

Разберем пример

Пример

```
class Point {  
    double x, y;  
public:  
    Point(double x = 0, double y = 0): x(x), y(y) {}  
    void setX(double _x) {  
        x = _x;  
    }  
};
```

Пример

```
class Point {  
    double x, y;  
public:  
    Point(double x = 0, double y = 0): x(x), y(y) {}  
    void setX(double _x) {  
        x = _x;  
    }  
};
```

Почему в конструкторе
параметры совпадают с полями
класса, а в сеттере - нет?

Пример

```
class Point {  
    double x, y;  
public:  
    Point(double x = 0, double y = 0): x(x), y(y) {}  
    void setX(double _x) {  
        x = _x;  
    }  
};
```

Что произойдет, если в сеттере
параметр тоже будет x?

Пример

```
class Point {  
    double x, y;  
public:  
    Point(double x = 0, double y = 0): x(x), y(y) {}  
    void setX(double x) {  
        x = x;  
    }  
};
```

Что произойдет, если в сеттере
параметр тоже будет x?

Указатель **this**

this - указатель, существующий в каждом методе класса и указывающий на тот объект, к которому применяется метод.

Пример

```
class Point {  
    double x, y;  
public:  
    Point(double x = 0, double y = 0): x(x), y(y) {}  
    void setX(double _x) {  
        x = _x;  
    }  
};  
...  
Point a, b;  
a.setX(1); // this указывает на объект a  
b.setX(-4); // this указывает на объект b
```

Использование this

```
class Point {  
    double x, y;  
    public:  
        Point(double x = 0, double y = 0): x(x), y(y) {}  
        void setX(double x) {  
            this->x = x;  
        }  
};  
...  
Point a, b;  
a.setX(1); // this указывает на объект a  
b.setX(-4); // this указывает на объект b
```

Почему ->?

**Можно ли обратиться к полям
через точку?**

Ещё один способ использовать `this`

Используя указатель `this` можно вернуть объект, к которому применялся метод, как результат работы метода.

Это позволит делать “цепочки” вызовов методов.

Пример

```
class Point {  
    double x, y;  
public:  
    Point(double x = 0, double y = 0): x(x), y(y) {}  
    void setX(double x) { this->x = x; }  
    Point& move(double x, double y) {  
        this->x = x; this->y = y;  
        return *this;  
    }  
    void print() {cout << x << " " << y;}  
};  
...  
Point a;  
a.move(1, 2).print();
```

Пример

```
class Point {  
    double x, y;  
public:  
    Point(double x = 0, double y = 0): x(x), y(y) {}  
    void setX(double x) { this->x = x; }  
    Point& move(double x, double y) {  
        this->x = x; this->y = y;  
        return *this;  
    }  
    void print() {cout << x << " " << y;}  
};  
...  
Point a;  
a.move(1, 2).print();
```

Важно! Результат должен быть ссылкой на объект, иначе результатом будет не сам объект, а его копия

Тест



10 минут



Разбор



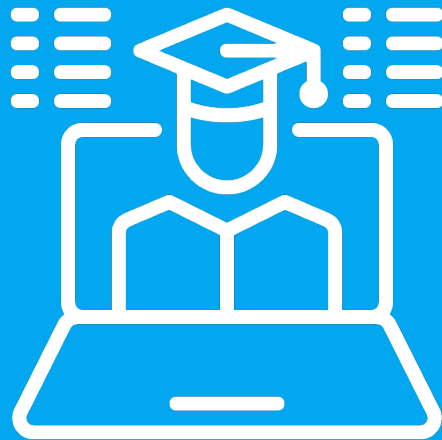
10 минут



Практика



35 минут



Итоги урока

- 1) Повторили указатели
- 2) Указатель `this`
- 3) Обращение к полям объекта через указатель
- 4) Результат метода - объект, к которому метод применялся
- 5) Цепочки вызовов методов