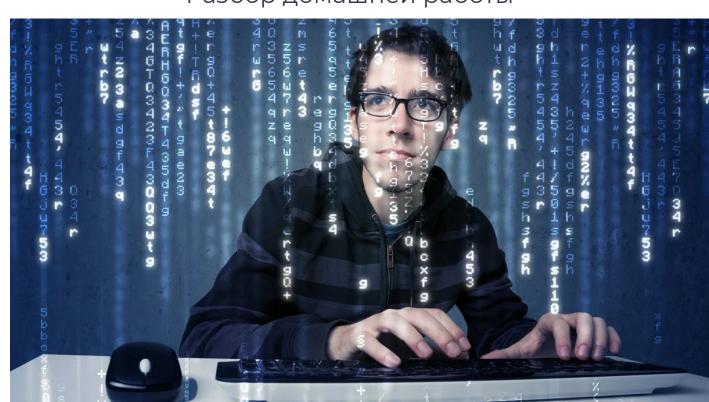




Разбор домашней работы



Конструкторы



Теория



15 минут





Конструктор

Конструктор - функция, запускаемая в момент создания объекта класса

Зачем могут понадобиться конструкторы?

Если создать объект класса, то какие значения будут храниться в его полях?

```
class Point {
    double x = 0;
    double y = 0;

public:
    void print() {
        cout << "(" << x << "; " << y << ")" << endl;
    }
};</pre>
```

```
class Point {
    double x = 0;
    double y = 0;

    public:
        void print() {
            cout << "(" << x << "; " << y << ")" << endl;
        }
};</pre>
```

```
class Point {
   double x = 0;
   double y = 0;
public:
   Point(int x, int y) {
        x = _x;
        y = y;
   void print() {
       cout << "(" << x << "; " << y << ")" << endl;
```



```
class Point {
                                               Тип возвращаемого
    double x = 0;
                                            значения не указывается
   double y = 0;
public:
    Point(int _x, int _y) {
        x = _x;
        y = y;
   void print() {
       cout << "(" << x << "; " << y << ")" << endl;
```

```
class Point {
   double x = 0;
   double y = 0;
public:
   Point(int x, int y) {
        x = _x;
        y = y;
   void print() {
       cout << "(" << x << "; " << y << ")" << endl;
Point p(1, 2); //Точка с координатами x=1 и y=2
```

```
class Point {
   double x = 0;
   double y = 0;
public:
   Point(int x, int y) {
                                            Ошибка компиляции
        x = _x;
        y = y;
   void print() {
       cout << "(" << x </"; " << y << ")" << endl;
Point p; // Точка с координатами x=0 и y=0?
```



Почему возникает ошибка компиляции в предыдущем примере?



Конструкторы

До того, как в классе реализован хотя бы один конструктор, в классе есть неявный конструктор без параметров, который ничего не делает, но он есть.

Если же в классе реализовать хотя бы один конструктор, то неявный конструктор из класса убирается, а значит создать объект без параметров не получится.

Как можно поступить в такой ситуации?

```
class Point {
   double x = 0, y = 0;
public:
   Point() {}
   Point(int x, int y) {
                                           Ошибки компиляции
       x = x;
                                                больше нет
       y = y;
   void print() {
      cout << "(" << x </"; " << y << ")" << endl;
Point p; // Точка с координатами x=0 и y=0
```



Сколько конструкторов может? быть в одном классе?



Практика



15 минут





Списки инициализации



Список инициализации

Когда запускается конструктор, сначала поля класса создаются либо с мусором, либо со значениями по умолчанию, если такие указаны, а после уже полям присваиваются новые значения. Это не так эффективно. Гораздо лучше было бы сразу создавать поля с нужными значениями.

Список инициализации позволяет инициализировать значения полей ещё до запуска тела конструктора, т.е. в момент создания этих самых полей

```
class Point {
    double x = 0, y = 0;
public:

    Point(int x, int y): x(x), y(y) {}
    void print() {
        cout << "(" << x << "; " << y << ")" << endl;
    }
};</pre>
```

```
class Point {
   double x = 0, y = 0;
public:
    Point(int x, int y): x(x), y(y) {}
    void print() {
       cout << "(" << x << "; \" << y << ")" << endl;
};
                                 Имена параметров и
                              полей класса теперь могут
                                  быть одинаковыми
```



Можно ли сделать и список инициализации, и тело конструктора?



Можно ли сделать и список инициализации, и тело конструктора?

Да

23 ====

Что в таком случае будет выполнено первым?



Что в таком случае будет выполнено первым?

Сначала выполнится список инициализаци

Ещё один пример использования конструктора

```
class Point {
    //Point из предыдущих примеров
};
int main() {
    Point p;
    //Что-то происходит
    p = Point(1, 2);
    //Присваиваем точке другую точку с координатами x=1 и y=2
    return 0;
```



Практика



40 минут





Разберем, как можно сделать прямоугольник, используя класс точки

Итоги урока

- 1) Что такое конструктор
- 2) Как создать конструктор
- 3) Что такое конструктор по умолчанию
- 4) Списки инициализации и как их создавать