



# 对象 构造器

## ✓ Class & Object

### ◆ 类是说明书，定义属性和行为

```
public class Dog {  
    String breed; // 成员变量  
    int age;      // 成员变量  
  
    public void bark() { // 成员方法  
        System.out.println("汪！");  
    }  
}
```

### ◆ 对象是类的具体实例：

```
Dog d = new Dog(); // 创建对象
```

## ✓ 构造器 Constructor

### ◆ 构造器是创建对象时自动调用的方法，用来初始化对象

A constructor in Java is a **special method** that is used to **initialise objects**.

The constructor is called when an object of a class is created.

It can be used to set initial values for object attributes

```

public class Main {
    int x;

    public Main(int y) {
        x = y;
    }

    public static void main(String[] args) {
        Main myObj = new Main(5);
        System.out.println(myObj.x);
    }
}

// Outputs 5

```

## !!! 构造器的运用

```

public class 类名 {          创建一个类
    数据类型 成员变量名1;
    数据类型 成员变量名2;

```

```

public 类名=构造器名 ( 数据类型 变量名1, 数据类型 变量名2 ){ 创建constructor
    成员变量名1 = 变量名1;
    成员变量名2 = 变量名2;
}

```

```

public static void main ( String[] args) {    程序入口
    类名作数据类型 新变量名 = new 类名/构造器名 (参数1👉变量名1, 参数2👉变量名2);

```

也就是说new出一个对象的时候将参数1赋值给变量名1，参数2赋值给变量名2  
然后变量名1赋值给了成员变量名1，变量名2赋值给了成员变量名2

这样，new出来的对象就是按照类作模版做出的object  
但是new出来的对象的成员变量是赋值进去的参数

```
public class Student {  
    String name;           🙌 成员变量 name  
  
    public Student(String name) {   🙌 构造器的参数也是 name  
        this.name = name;         🙌 this == 类  
                                   📌 this.name == 类.成员变量  
                                   类.成员变量 = 构造器的参数  
    }  
    也就是 把参数 name 的值赋给当前对象的成员变量 name  
}
```

- `this.name` → 是这个对象的名字属性
- `name` → 是传进来的参数
- 所以我们写 `this.name = name;`，意思是把参数 `name` 的值，赋值给这个对象的属性 `name`

```
public Dog(String b, int a) {   Dog == 类名 == 构造器的方法名  
    breed = b;  
    age = a;  
}
```

- 构造器没有返回值（连 `void` 都不写）
- 名字必须和类名一样
- 如果你没有声明任何constructor, 编译器会自动生成一个构造器

## ✅ 为什么「必须通过构造器才能创建实例（对象）」？

因为 Java 中的对象是通过 `new` 操作符调用构造器（`constructor`）方法创建的，这个过程完成两件事：

1. 在内存中分配空间 ( new )
2. 执行初始化操作 (构造器)

## ✓ 没写构造器怎么办？

Java 会默认提供一个「无参构造器」：

```
public class Cat {  
    String name;  
}  
// 可以直接写：new Cat();
```

⚠ 一旦你写了自己的构造器（带参数的），Java 就不会自动加默认构造器了

```
public class Main {  
    public static void main(String[] args) {  
  
        Cat cat = new Cat();    // ① 创建一个 Cat 对象，调用无参构造器  
        cat.name = "张三";    // ② 手动赋值给成员变量 name  
  
        Cat cat2 = new Cat("李四"); // ✓ ③ 前提是你写了带参构造器  
    }  
}
```