

类和对象

面向过程 vs 面向对象 (OOP, Object Oriented Programming)

类：抽象概念（特性，功能），不指具体事物

对象：类的特性有具体值，实现类描述功能，具体的事物，每一个都不一样

面向对象特征：封装，继承，多态

类的定义

```
[修饰符] class 类名{
    零个或多个属性/成员变量

    零个或多个构造器

    零个或多个方法
}

package javaoopday01;

//用class关键字定义类，文件名和类名相同的，首字母大写，符合标识符命名规则
public class Student {
    /*
        属性（成员变量/字段）。
        直接写到类下面的变量，类型可以是基本类型或引用类型
        比局部变量多了修饰符，一般情况用private
        可以不用初始化，因为它跟数组一样，有默认的初始值
    */
    private int age;
    private String name;

    /*
        * 构造器
        * 当使用new关键字实例化对象时，就会调用构造器
        * 作用：时初始化属性的值
        * 特点：没有返回类型；方法名跟类名相同；修饰符一般为public；参数可以有0个或多个
    */
    public Student(int i, String s) {
        age = i;
        name = s;
    }

    /*
        * 方法时类实现行为的代码
        * 特点：修饰符；返回类型；方法名符合标识符命名规则，首字母小写；参数可以有0个或多个；方法
        体实现行为功能的代码
    */
    public void study() {
        System.out.println("学习");
    }
}
```

```

    public void sport() {
        System.out.println("运动");
    }

    public void info() {
        System.out.println("姓名: " + name + ", 年龄: " + age);
    }
}

```

构造器

构造器是类实例化（new xx()）成对象的时候调用的特殊方法。目的是用于初始化属性

每个类都有构造器，如果代码没有写，Java会给它生成无参数的构造器。如果代码添加了构造器，那个系统生成的默认构造器就不复存在。

构造器特点：无返回类型（连void都不用），方法名跟类名相同，参数可以有零或多个

构造器里特殊方法调用：

- super () 调用父类的构造方法，代码只能放在构造器的第一行
- this () 构造方法里面调用另一个构造方法，用this()，调用代码只能放在构造器首行

属性

[修饰符] 类型 属性名 [= 属性值]

```

private int a;
private String b;

```

当在方法里出现跟属性名同名的局部变量时，如果直接写变量名都是指局部变量，要在变量前this.才能访问属性

```

this.field2 = field2;

```

属性可以不用初始化，它默认值：整型默认是0；浮点型默认是0.0，布尔型默认是false；char默认是'\u0000'；应用类型默认null

修饰符控制属性或方法的访问权限（能够被哪些类访问），修饰符一共有四种：private，default，protected，public

	private (私有的)	default (默认的)	protected (受保护的)	public (公共的)
同一个类	√	√	√	√
同一个包		√	√	√
子类中			√	√
其它类				√

为了保护数据，更好实现面向对象的封装特性，属性的修饰符一般都设置private，修改属性的值通过方法去实现。

如果一个类有属性，每个属性有get, set方法，这个类称为Java Bean（实体类），一般用于跟数据库表映射。

属性除了四种修饰符之一外，还可以与static和final之一（或者二者）一起使用，public static final 这样修饰的属性通常用于全局变量：

- static 静态属性（或叫类的属性），可以直接用类名.属性名的方式访问，也可以通过对象访问；所有对象共享同一个属性值
- final final修饰的属性不能再次赋值，必须在定义的时候赋初始值

方法

```
[修饰符] 返回类型 方法名(参数列表){  
    //方法代码  
}
```

方法的修饰符也是可以用private, default, protected, public之一，访问权限跟属性一样。还可以与static 或 final和abstract之一组合使用。

static方法可以直接用类名访问，多用于工具类的方法。静态方法里面不能调用非静态方法。

方法必须要设置返回类型，如果没有返回返回值，就用void。有返回值的方法代码里有return语句，并且return 类型要个返回类型一致。

方法的参数可以有零个或多个，多个参数用逗号分隔，参数的定义格式“类型 参数名”。在方法定义时参数叫形式参数（简称形参），参数可以任意起名，在方法里面使用参数跟形参名字一样；在调用方法的时候给出参数叫实际参数（简称实参），实参的类型必须跟形参类型一致。

可变长度参数定义方式为类型后面加...，可变长度参数只能放在形参最后一个

//形参类型后面加三个点， 表示可变长度参数。可变长度参数必须放在形参的最后一个

```
public void printBook(String... book) {  
    for(String bookName : book) {  
        System.out.println(bookName);  
    }  
}
```

methodInvoke.printBook("Java","Sql","C", "Python");//调用可变长度参数方法时，实参可以传任意个数

methodInvoke.printBook();//调用可变长度参数方法时，实参可以不传

//下面时错误的形参定义,因为可变参数不是最后一个参数

```
public void printBook(String... book, String name) {
```

今日作业

1. 什么是类？什么是对象？类和对象的关系？
2. 面向对象的三大特征？
3. 什么是构造器，它的作用是什么？
4. 编程。定义一个类Rect，设置属性x, y, width, height, 添加无参及有参构造器
5. 编程。定义订单实体类Orders, 有属性odrId, odrPrice, odrNum, odrAccount, 生成get和set方法
6. 扩展编程。写一个图书管理类，有学生学号信息，有图书书名。行为：验证学生学号；学生给出书名，查找这本。