/\*

\* 构造方法：给对象数据进行初始化

\* 1、方法名与类名相同

\* 2、没有返回值类型，连void都没有

\* 3、没有具体的返回值

\* 重载：方法名与类名相同， 无返回值，参数类型不同

\*{ 如果不给构造方法系统将提供无参构造方法，

\* 但是只要给定了构造方法系统将不提供无参的构造方法，

\* 如果还要使用无参构造方法必须自己给出 }

\*/

**class** Stu {

String brand;

**int** price;

Stu(String brand,**int** price) {

**this**.brand=brand;

**this**.price=price;

}

Stu(String brand){

**this**.brand = brand;

}

**public** **void** getBrand() {

System.***out***.println(brand);

}

**public** **void** get() {

System.***out***.println(brand+" "+price);

}

}

**public** **class** GouZaoDemo1 {

**public** **static** **void** main(String[] args) {

Stu st = **new** Stu("三星",2005);

st.get();

//

System.***out***.println("----------");

Stu s2 = **new** Stu("华为");

s2.getBrand();

}

}

\*static (可以修饰成员变量以及方法):

\* 修饰的数据被所有对象共享

\* 随着类的加载而加载

\* 使用情景：如果某个成员变量是被所有对象共享的，就应该定义为静态的

\* (可以通过类名调用,也可以通过对象名调用）

\* 注意事项：在静态方法中是没有this关键字的

\* （静态方法只能访问静态成员方法和静态的成员变量）

（静态比对象先存在）

静态方法：只可以访问静态变量

非静态方法：可以访问非静态以及静态变量,以及方法

\*/

**class** Text {

**int** num = 10;

**static** **int** *num2* = 20;

//非静态方法：可以访问非静态以及静态变量,以及方法

**public** **void** printNum() {

System.***out***.println(num);

System.***out***.println(*num2*);

}

//静态方法：只可以访问静态变量

**public** **static** **void** printNum2() {

System.***out***.println(*num2*);

}

}

**public** **class** StaticDemo {

**public** **static** **void** main(String[] args) {

Text te = **new** Text();

te.printNum();

te.*printNum2*();

}

}

**package** ch7;

//变量

/\*

\* 成员变量：在类中方法外（有默认值）

\* 局部变量：在方法定义中或者在方法声明上（没有默认值必须初始化定义后才能使用）

当成员变量和局部变量一样时采用就近原则

\*/

/\*

\* 基本类型： 基本类型：

引用类型：（类以及数组）必须创建相应类的对象

匿名对象：没有名字的对象

应用场景：调用方法：仅仅只是调用一次的时候

匿名对象作为实际参数的使用

\*/

**class** Student {

**public** **void** show() {

System.***out***.println("study");

}

}

**class** StudentDemo {

//如果看到了一个方法的形式参数是一个类类型（引用类型），需要创建该类的对象

**public** **void** method(Student s) {//调用的时候是将地址值传送到这里

s.show();

}

}

**public** **class** VariableDemo1 {

**public** **static** **void** main(String[] args) {

//形式参数是引用类型的调用

StudentDemo sc = **new** StudentDemo();

//必须创建类类型的对象

Student s = **new** Student();

sc.method(s);//把s的地址传送到了这里

//类名字的调用

s.show();

//匿名对象

**new** Student();

//匿名对象调用方法

**new** Student().show();

**new** Student().show();//创建了一个与上面不是同一个的新对象

//

StudentDemo ss = **new** StudentDemo();

//匿名对象可以作为实际参数使用

ss.method(**new** Student());//创建了student的匿名对象

}

}