

一. 创建对象格式

```
1 类名 对象名=new 构造方法();  
2 Girl girl=new Girl();  
3 girl.sayHi();
```

```
1 Girl girl1=new Girl("Lily","女");  
2 girl1.sayHi();
```

二. 静态属性(类属性)和非静态属性(实例属性)的区别

1. 非静态属性通过对象获取, 静态属性通过类和对象获取, 推荐使用类
2. 每个对象都有属于自己的非静态属性; 多个对象公用同一个静态属性

```
1 System.out.println(girl.name);  
2 System.out.println(Girl.age);
```

```
//为属性赋默认值  
//1. 方式1,  
public String name="小华";  
public String gender="女";  
  
//静态属性  
//静态属性  
public static int age=10;  
  
//方式2:  
//构造方法:优先级更高  
//特点:1. 没返回值数值类型,  
// 2. 方法名和类名保持一致  
// 3. 当类中没有构造方法是, 系统会自动生成一个无参构造方法4. 配合new. 用于创建对象  
//5. 用于初始化属性的值  
public Girl() {  
    name="飞飞";  
    gender="女";  
}
```

三. 静态方法(类方法)和非静态方法的区别

1. 非静态方法通过对象调用, 静态方法通过对象调用, 推荐使用类

2. 在非静态方法中可以调用在非静态方法和静态方法, 在静态方法只能调用静态方法

3. 在非静态方法中可以调用在非静态属性和静态属性, 在静态方法只能使用静态方法

4. this只能在非静态方法中使用

```
1 girl.goShopping();
2 girl.makeup();
```

```
//静态属性
//静态属性
public static int age=10;

//方式2:
    //构造方法:优先级更高
    //特点:1. 没返回值数值类型,
// 2. 方法名和类名保持一致
// 3. 当类中没有构造方法是, 系统会自动生成一个无参构造方法4. 配合new. 用于创建对象
//5. 用于初始化属性的值
public Girl() {
    name="飞飞";
    gender="女";
}

public Girl(String name,String gender){
    //使用this区分属性和参数
    this.name=name;
    this.gender=gender;
}
```

```
public void playGame() {  
    System.out.println("玩游戏");  
    System.out.println(gender);  
    System.out.println(age);  
    //this:相当于调用该方法的对象  
}
```

```
public static void learn() { System.out.println("学习"); }  
public void goShopping() { System.out.println("逛街"); }
```

```
public static void makeup() {  
    learn();  
  
    System.out.println("化妆");  
    //System.out.println(gender);  
    System.out.println(age);  
}
```

```
public void sayHi() { System.out.printf("我是你的%s朋友, %s!\n", gender, name); }
```

```
}
```