继承

1. 继承
2. 什么是继承

所谓继承就是用一个子类(派生类)继承父类(基类|超类).

1. 继承的作用:代码的重用性（体现在设计上）
2. 使用extends实现继承

public 子类名 extends 父类名{

一系列的属性和方法…..

}

注意:

1. 父类私有成员、构造方法不能被继承
2. 继承只能实现单继承(单根性)、无法实现多重继承(借助接口实现多继承)
3. super关键字的用法
4. 在子类中使用super关键字访问父类成员

super.属性名|方法名(参数);

1. 在子类构造方法中使用super关键字调用父类构造

super(构造方法的参数列表);

1. (重点理解)实例化子类时构造方法的调用过程

当实例化子在时，首选匹配子类的构造,在子类的构造方法的第一行使用super(默认)关键调用父类构造，接着执行子类构造方法内容.

1. 方法重写(注意:只发生在继承中)
2. 作用:用于实现多态
3. 实现方法重写的步骤:

2.1第一步:定义一个父类添加可能被重写的方法

|  |
| --- |
| **public class** Parent { *//父类  //父亲走路的行为* **public void** walk(){  System.***out***.println(**"父亲走路正常:一左一右一左一右一左一右一左一右。。"**);  } } |

* 1. 第二步:定义子类去继父类，并重写父类的方法

|  |
| --- |
| **public class** Son **extends** Parent { //*称为重写方法*  *//父亲走路的行为*  **public void** walk(){  System.***out***.println(**"儿子的走路行为....走路不正常。。。。。。。"**);  } } |

注意:什么是重写方法:在子类中定义一个与父类具有相同签名的方法称为重写方法.

2.3发生重写:

|  |
| --- |
| *//发生重写:用子类的方法去重写父类的方法 // 父类名 对象名=new 子类名(); 父类等于子类对象* Parent parent=**new** Son();  parent.walk(); |