继承

课程目标:

1. 理解继承和实现 掌握
2. Super关键字
3. 理解子类构造方法的调用过程 重点难点
4. 方法重写 掌握
5. 继承
6. 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)

示例:

|  |
| --- |
| *//人类* **public class** Person {   **public** String **name**;  **public int age**;  *//显示* **public void** show(){  System.***out***.println(**"姓名:"**+**this**.**name**);  System.***out***.println(**"年龄:"**+**this**.**age**);  } }  *//学生类* **public class** Student **extends** Person{  **public int xh**;  **public void** show(){  System.***out***.println(**"学号:"**+**this**.**xh**);  System.***out***.println(**"姓名:"**+**this**.**name**);  System.***out***.println(**"年龄:"**+**this**.**age**);  } }  *//员工类* **public class** Emp **extends** Person{  **public int gh**;  **public void** show(){  System.***out***.println(**"工号:"**+**this**.**gh**);  System.***out***.println(**"姓名:"**+**this**.**name**);  System.***out***.println(**"年龄:"**+**this**.**age**);  } }  **public static void** main(String[] args) {  *//创建学生对象  Student stu=new Student();  stu.xh=101;  stu.name="张三";  stu.age=21;  stu.show(); }* |

1. super关键字的用法
2. 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，首先匹配子类的构造,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.

示例:

|  |
| --- |
| *//人类* **public class** Person {  **public** String **name**;  **public int age**;  **public** Person(){  System.***out***.println(**"我是父类构造方法。。。。什么时候执行"**);  }  **public** Person(String name,**int** age){  **this**.**name**=name;  **this**.**age**=age;  }  *//显示* **public void** show(){  System.***out***.println(**"姓名:"**+**this**.**name**);  System.***out***.println(**"年龄:"**+**this**.**age**);  } }  *//学生类* **public class** Student **extends** Person{  **public** Student(){  *//第一行:采用super关键字调用父类构造 ,默认就是super()* **super**();  System.***out***.println(**"我是子类构造方法。。。。什么时候执行"**);   }  *//定义带参构造进行学生属性的初始化* **public** Student(**int** xh,String name,**int** age)  {  **super**(name,age); *//调用带参的父类构造* **this**.**xh**=xh;  *//this.name=name;  //this.age=age;* }  **public int xh**;  **public void** show(){  System.***out***.println(**"学号:"**+**this**.**xh**);  System.***out***.println(**"姓名:"**+**this**.**name**);  System.***out***.println(**"年龄:"**+**this**.**age**);  } }  主函数：  *//创建学生对象时并对属性初始化---->利用构造* Student stu=**new** Student(101,**"张三"**,21); stu.show(); |

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(); |

1. 使用static关键字
2. 作用:定义静态成员，与对象无关，归类所有

定义静态代码块

1. 定义静态成员

|  |
| --- |
| **public class** MathTool {  *//1.定义静态属性* **public static int** *r*;   *//2.定义静态方法* **public static void** getMj(){  **double** res=3.14\**r*\**r*;  System.***out***.println(**"面积是:"**+res);  } } |

1. 使用静态成员

|  |
| --- |
| **public static void** main(String[] args) {  *//使用静态成员(归类所有)  //1.访问静态属性* MathTool.*r*=2;  *//2.访问静态方法* MathTool.*getMj*(); } |

1. 使用static关键字定义静态代码块

|  |
| --- |
| *//静态代码块 //注意:静态代码块是当类被加载时(加载到JVM)执行* **static** {  System.***out***.println(**"我是静态代码块,我什么时候执行"**); } |