**一：返回值**

|  |
| --- |
| //返回值举例  **class** Test0320{  **public** **static** **void** main(String []args){  **int** x=Test0320.*run*(); //执行run()方法后返回值赋值给x  Test0320.*runResult*(x);//调用runResult()方法，传入参数  }  **public** **static** **int** run(){  **int** a;  a=2+4;  **return** a;  }  **public** **static** **void** runResult(**int** y){  System.***out***.println(y);  }  } |

**二：简单封装**

代码一：

|  |
| --- |
| **package** com.zhiyou.test;  **class** Test1{    **private** **int** chang;  **private** **int** kuan;    **public** **void** mianji(){    System.***out***.println("面积为："+chang\*kuan);  }  }  **class** Test2 {    **public** **static** **void** main(String []args){  Test1 t1=**new** Test1();  t1.chang=6; //报错，因为int型变量chang为私有的，不允许其他类调用  t1.kuan=4; //报错，因为int型变量kuan为私有的，不允许其他类调用  t1.mianji();  }  } |

**将上面代码变形为：**

|  |
| --- |
| **package** com.zhiyou.test;  **class** Test1{    **private** **int** chang;  **private** **int** kuan;    **public** **void** setChang(**int** a){  a=chang; //通过给实参给私有的chang赋值  }  **public** **void** setKuan(**int** b){  b=kuan; //通过给实参给私有的kuan赋值  }  **public** **void** mianji(){    System.***out***.println("面积为："+chang\*kuan);  }  }  **class** Test2 {    **public** **static** **void** main(String []args){  Test1 t1=**new** Test1();  t1.setChang(6);  t1.setKuan(4);  t1.mianji();  }  } |

代码二：

|  |
| --- |
| **package** com.zhiyou.test;  //将所需属性单独封装成一个类  **class** User{  **private** **int** chang;  **private** **int** kuan;  **public** **int** getChang() {  **return** chang;  }  **public** **void** setChang(**int** chang) {  **this**.chang = chang;  }  **public** **int** getKuan() {  **return** kuan;  }  **public** **void** setKuan(**int** kuan) {  **this**.kuan = kuan;  }  } |

|  |
| --- |
| **package** com.zhiyou.test;  **public** **class** Test {  **public** **static** **void** main(String[] args) {  User u=**new** User();  u.setChang(6);  u.setKuan(4);  Test0320\_1.*mianji*(u);  }  **public** **static** **void** mianji(User user){  System.***out***.println("面积为："+user.getChang()\*user.getKuan());  }  } |

**三：继承**

继承 继承类所有的成员方法和成员变量但是不包括private 修饰的成员变量和成员方法，可以累加继承，只允许单向继承，不允许多向继承（也就是不允许一个类同时继承两个类）

|  |
| --- |
| **package** com.zhiyou.test;  **class** person{    private String name;  **int** age;    **public** **void** setName(String s){  s=name;  }  **public** **void** chi(){  System.***out***.println("吃吃吃！！！");  }  }  **class** man **extends** person{    **public** **void** work(){  **super**.chi();  //报错，因为父类person中name是私有的，不能被继承  //将person中的name标识去掉即可执行成功！！！  System.***out***.println("男人："+name+" 年龄："+age+"good good working!!!");  }  }  **class** xiaoman **extends** man{    **public** **void** study(){  **super**.work();  //报错，因为父类man继承的person中name是私有的，不能被继承  //将person中的name标识去掉即可执行成功！！！  System.***out***.println("学生："+name+" 年龄："+age+"good good studay!!!");  }  }  **class** woman **extends** person{  **public** **void** shopping(){  //报错，因为父类person中name是私有的，不能被继承  //将person中的name标识去掉即可执行成功！！！  System.***out***.println("女人："+name+" 年龄："+age+" good good shopping!!!");  }  }  **class** Test{    **public** **static** **void** main(String []args){  man m=**new** man();  m.setName("zhangsan");  m.age=20;  m.work();  xiaoman xm=**new** xiaoman();  xm.study();  }  } |

**四：多态（重写和重载）**

**《1》重写**

|  |
| --- |
| **package** com.zhiyou.test;  **class** person{  String name;  **int** age;    **public** **void** run(){  System.***out***.println("这是父类的run方法！！！");  }  }  **class** man **extends** person{    **public** **void** run(){  **super**.run();//利用super关键字调用父类的run()方法  System.***out***.println("这是子类的run方法！！！");  }  }  **class** Test{  **public** **static** **void** main(String []args){  person p=**new** man();//引用子类的实例  p.run();  }  } |

**《2》重载**

|  |
| --- |
| **package** com.zhiyou.test;  **class** Test0320\_1{  **public** **static** **void** main(String []args){    Test0320\_1.*run*();  Test0320\_1.*run*("带有一个参数");  Test0320\_1.*run*("带有两个参数", 8);  }    **public** **static** **void** run(){    System.***out***.println("这是run方法");  }    **public** **static** **void** run(String s){    System.***out***.println("这是重载的run方法："+s);  }  **public** **static** String run(String s,**int** a){    System.***out***.println("这是重载run方法"+s+"..."+a);  **return** "hehe";  }  } |

**五：构造方法**

|  |
| --- |
| **package** com.zhiyou.test;  /\*\*  \*1.初始化  \*2.类在默认情况下会有一个跟类同名的且无参的构造函数，如果你创建一个新的构造函数的话  \*默认的构造函数就没有了  \*3.构造函数在一个类中可以有多个  \*4.当对象在声明过程中会根据参数列表进行选择构造函数，如果没有对应的构造函数，则会实例化失败  \*5.子类不能继承父类的构造函数，但是在new子类的时候，会先去读取父类默认构造函数，如果父类的  \*构造函数重新编写过，那么子类也应该重新编写构造函数  \*/  **class** person{    person(){  //System.out.println("我是构造函数的输出语句！！！");  }  }  **public** **class** Test0320\_1{    **public** **static** **void** main(String []args){  person p=**new** person();  }  } |

|  |
| --- |
| **class** person{  person(){  System.***out***.println("我是person类构造函数的输出语句！！！");  }  person(String s){  System.***out***.println("我是person类带有参数的构造函数输出语句！！！");  }  }  **class** man **extends** person{    **public** **void** run(){    System.***out***.println("我是man类的run方法！！！");  }  }  **public** **class** Test0320\_1{    **public** **static** **void** main(String []args){  person p=**new** person();  person p1=**new** person("zhangsan");  //person p2=new person("zhangsan","lisi");//报错  man m=**new** man();  m.run();  }  } |

**六：static静态**

|  |
| --- |
| **package** com.zhiyou.test;  /\*\*  \* **static 修饰变量的特点**：  \* 1.使用static修饰的变量称为静态变量，存放在内存共享区域内，被所有对象所共享，只加载一次  \* 2.如果有一个对象将静态变量值修改，那么其他对象访问的值为修改过后的值  \* 3.static可以用来修饰成员变量，不能用来修饰局部变量  \*/  **class** man{    String name="zhangsan";    **public** **void** show(){  System.***out***.println(name);  }  }  **public** **class** Test0320\_1{    **public** **static** **void** main(String []args){  man m=**new** man();  m.name="lisi";  m.show();    man m1=**new** man();  m1.show();  }  } |

|  |
| --- |
| **class** man{    **static** String *name*="zhangsan";    **public** **void** show(){  System.***out***.println(*name*);  }  }  **public** **class** Test0320\_1{    **public** **static** **void** main(String []args){  man m=**new** man();  m.*name*="lisi";  m.show();    man m1=**new** man();  m1.show();  }  } |

|  |
| --- |
| **package** com.zhiyou.test;  /\*\*  \* static修饰成员方法  \* 1.使用static 修饰的成员方法被称为类方法，被所有对象共享，可以直接进行引用（类名.方法名()）  \* 2.静态方法只能本类中的静态成员，不能引用其他非静态成员  \* 3.其他非静态的成员方法可以直接调用静态方法  \* 4.静态方法不能直接使用this，super等关键字  \*/  **class** man{    **static** String *name*="zhangsan";    **public** **static** **void** show(){  System.***out***.println(*name*);  }    **public** **void** show2(){  *show*();  System.***out***.println("show2");  }  }  **public** **class** Test0320\_1{    **public** **static** **void** main(String []args){  man m1=**new** man();  m1.show2();  }  } |