**每日作业卷**

**就业班JavaSE第1天 基础回顾与继承**

# 关卡1

## 训练案例1

### 训练考核知识点

this关键字

### 训练描述

请阐述你对this关键字的理解

### 操作步骤描述

1. 请说this一般出现在哪里?

2. 请说this代表什么?

3. 请说出this是谁?

1. 类属性的get，set方法

2. 当前对象

3. 当前对象

## 训练案例2

### 训练考核知识点

类名作为参数类型和返回值类型

### 训练描述

请阐述你对类名作为参数类型和返回值类型的理解

### 操作步骤描述

1. 请说出类名是谁的数据类型
2. 请说出当类名做为一个方法的参数类型的时候,调用这个方法的时候,需要传入的是什么?
3. 请说出当一个方法的返回值类型是类名时候,调用这个方法,该方法返回的值是什么?

1. 对象

2. 用这个类名创建的对象

3. 这个类对象

## 训练案例3

### 训练考核知识点

匿名对象

### 训练描述

请阐述你对匿名对象的理解

### 操作步骤描述

1. 请说出什么是匿名对象

2. 请说出匿名对象和有名对象的区别

3. 请说出匿名对象的使用场景

1. 没有名字的对象

2. 匿名对象运行一次，直接就被回收掉了

3. 调用方法,仅仅只调用一次的时候

## 训练案例4

### 训练考核知识点

组合关系

### 训练描述

请阐述你对组合关系的理解

### 操作步骤描述

1. 请说出什么是组合关系?

2. 请举出一个组合关系的例子

1. 当一个自定义类型A的成员变量的数据类型是自定义类型B时

2. 人Person与宠物Pet就是一种组合的关系

## 训练案例5

### 训练考核知识点

继承

### 训练描述

请阐述你对继承的理解

### 操作步骤描述

1. 请说出什么是继承?

2. 请说出子类访问父类成员变量的特点 ?

3. 请说出子类调用父类方法的特点 ?

4. 请说出继承的特点是什么?

1. 继承是代码复用的重要方式

2. 子类可以访问父类的public成员

3. 可以重写父类方法

4. 子类属于父类的一种

## 训练案例1

### 训练考核知识点

标准类的定义和使用

### 训练描述

请使用代码描述： 2岁的棕色加菲猫,在抓老鼠

### 操作步骤描述

1. 定义Cat类
   1. 成员变量 名称(name), 颜色(color),年龄(age)
   2. 成员方法 抓老鼠(catchMouse())
   3. 提供空参和带参的构造方法
   4. 提供setter和getter方法

2. 创建测试类Test01

1. 提供main方法
2. 在main方法中
   1. 创建Cat对象,并把名称设置为加菲,颜色设置为棕色,年龄设置为2
   2. 调用Cat对象的catchMouse()方法,打印格式如下:

2岁的棕色加菲猫,在抓老鼠

**package** com.day10.test1;

**public** **class** Cat {

**private** String name;

**private** String color;

**private** String age;

**public** **void** catchMouse() {

System.***out***.println("在抓老鼠");

}

**public** Cat() {

**super**();

}

**public** Cat(String name, String color, String age) {

**super**();

**this**.name = name;

**this**.color = color;

**this**.age = age;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getColor() {

**return** color;

}

**public** **void** setColor(String color) {

**this**.color = color;

}

**public** String getAge() {

**return** age;

}

**public** **void** setAge(String age) {

**this**.age = age;

}

}

**package** com.day10.test1;

**public** **class** Test01 {

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

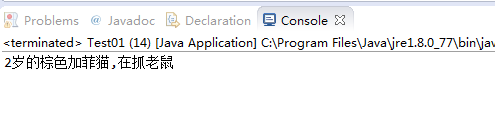
Cat cat = **new** Cat("加菲", "棕色", "2");

System.***out***.print(cat.getAge() + "岁的" + cat.getColor() + cat.getName() + "猫,");

cat.catchMouse();

}

}



## 训练案例7

### 训练考核知识点

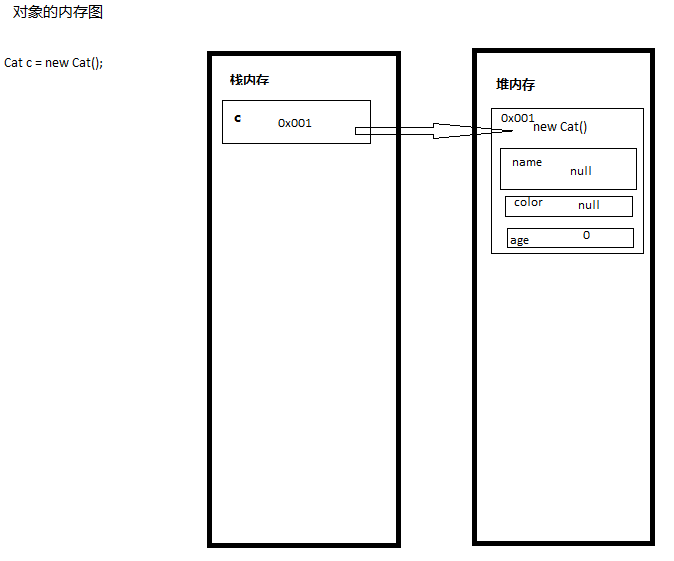
对象的内存图

### 训练描述

请绘制Cat c = new Cat(); 这句代码的内存图,说明:Cat类是第6题的Cat类.

### 操作步骤描述

1. 打开画图软件
2. 在画布的左上角写上: 对象的内存图
3. 在左边写上: Cat c = new Cat();
4. 在这句代码右边,绘制大长方形方块,在左上角标注 栈内存
5. 在这个长方形方块的右侧,在绘制一个同样大小的方块标注为: 堆内存
6. 在栈内存的方块内部,绘制一个小方块, 在左上角标注为c; 表示为局部变量c开辟的存储空间
7. 在堆内存的方块内部,绘制一个稍微大一点方块,用于描述创建对象分配的存储空间,在这个方块,上面写new Cat() 表示创建对象分配的存储空间
8. 在这个对象的方块中再绘制一个小方块,左上角标注为 name,表示为成员变量name分配的存储空间,小方法内部写上null,表示名称的默认值是null.
9. 再在对象的方块中绘制一个小方块,左上角标注为color,表示为成员变量color分配的存储空间,方块内部写上null,表示颜色的默认值为null
10. 再在对象的方块内部绘制一个小方块,左上角标注为age,表示为成员变量age分配的存储空间,方块内部写上0,表示age的默认值是0
11. 在对象方块的左上角,标注0x001, 表示该对象的内存地址
12. 在栈内存中,代表局部变量c的小方块中写上0x001,表示局部变量c中存储是对象的地址;
13. 绘制一条从栈内存局部变量c中0x001到堆内存对象的0x001带有箭头的线,表示栈内存的变量,指向堆内存的对象



# 关卡2

## 训练案例1

### 训练考核知识点

类名作为参数类型

### 训练描述

请使用代码描述: 18岁的景甜在使用iPhone手机玩游戏

### 操作步骤描述

1. 定义手机类(Phone)
   * 1. 成员变量: 品牌(brand)
     2. 成员方法: 玩游戏(void playGame())
     3. 提供空参和带参构造方法
     4. 提供setXxx和getXxx方法
2. 定义人类(Person)
   * 1. 成员变量: 姓名(name),年龄(age)
     2. 成员方法: 使用手机玩游戏 (usePhone(Phone p))
     3. 提供空参和带参构造方法
     4. 提供setXxx和getXxx方法
3. 定义Test类
   1. 提供main方法
   2. 在main方法中
      1. 创建手机对象,并把手机品牌赋值为iPhone
      2. 创建Person对象,并把名称赋值为景甜,年龄赋值为18
      3. 调用Person对象的,usePhone方法,传入手机对象
      4. 打印格式如下:

18岁的景甜在使用iPhone手机玩游戏

**package** com.day10.test2;

**public** **class** Phone {

**private** String brand;

**public** **void** playGame() {

}

**public** Phone() {

**super**();

}

**public** Phone(String brand) {

**super**();

**this**.brand = brand;

}

**public** String getBrand() {

**return** brand;

}

**public** **void** setBrand(String brand) {

**this**.brand = brand;

}

}

**package** com.day10.test2;

**public** **class** Person {

**private** String name;

**private** String age;

**public** **void** usePhone(Phone p) {

System.***out***.println(**this**.age + "的" + **this**.name + "在使用" + p.getBrand() + "手机玩游戏");

}

**public** Person() {

**super**();

}

**public** Person(String name, String age) {

**super**();

**this**.name = name;

**this**.age = age;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getAge() {

**return** age;

}

**public** **void** setAge(String age) {

**this**.age = age;

}

}

**package** com.day10.test2;

**public** **class** Test {

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

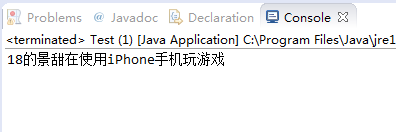
Phone phone = **new** Phone("iPhone");

Person person = **new** Person("景甜", "18");

person.usePhone(phone);

}

}



## 训练案例2

### 训练考核知识点

匿名对象

### 训练描述

请用代码描述: 周杰伦正在演唱菊花台歌曲(使用匿名对象)

### 操作步骤描述

1. 定义Person类
   1. 属性: 姓名(name)
   2. 行为: 唱歌(void sing(String song))
   3. 要求： 提供带参构造方法
2. 定义测试类Test
   1. 提供main方法
   2. 在main方法中
      1. 使用带参构造创建Person对象,把name赋值为周杰伦
      2. 使用上面创建的匿名对象调用唱歌的方法;唱歌的方法输出格式为

周杰伦正在演唱菊花台歌曲

**package** com.day10.test3;

**public** **class** Person {

**private** String name;

**public** **void** sing(String song) {

System.***out***.println(**this**.name + "正在演唱" + song + "歌曲");

}

**public** Person() {

**super**();

}

**public** Person(String name) {

**super**();

**this**.name = name;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

}

**package** com.day10.test3;

**public** **class** Test {

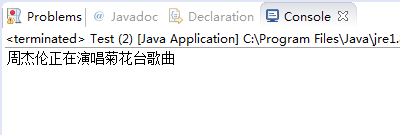
**public** **static** **void** main(String[] args) {

Person person = **new** Person("周杰伦");

person.sing("菊花台");

}

}



## 训练案例3

### 训练考核知识点

匿名对象调用方法，匿名对象作为实际参数,类名的作为参数类型

### 训练描述有

请用代码描述： 古力娜扎正在看择天记小说(使用匿名对象)

### 操作步骤描述

1. 定义书类(Book)
   1. 属性: 名称(name)
   2. 要求:
      1. 提供带参构造,setXxx与getXxx方法
2. 定义人类(Person)
   1. 属性: 名称(name)
   2. 行为: 看书: readBook(Book book)
   3. 要求: 提供带参构造
3. 定义测试类Test
   1. 提供main方法
   2. 在main方法中
      1. 使用带参构造创建Person对象,name初始化为古力娜扎
      2. 使用该匿名对象,调用readBook(Book book) 方法，使用Book的带参构造创建Book对象,把名称初始化为择天记,把Book的匿名对象作为参数传入readBook(Book book)方法
   3. 输出格式
      1. 古力娜扎正在看择天记小说

**package** com.day10.test4;

**public** **class** Book {

**private** String name;

**public** Book() {

**super**();

}

**public** Book(String name) {

**super**();

**this**.name = name;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

}

**package** com.day10.test4;

**public** **class** Person {

**private** String name;

**public** **void** readBook(Book book) {

System.***out***.println(**this**.name + "正在看" + book.getName() + "小说");

}

**public** Person() {

**super**();

}

**public** Person(String name) {

**super**();

**this**.name = name;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

}

**package** com.day10.test4;

**public** **class** Test {

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

Person person = **new** Person("古力娜扎");

Book book = **new** Book("择天记");

person.readBook(book);

}

}



## 训练案例4

### 训练考核知识点

匿名对象作为返回值,类名作为返回值类型

### 训练描述有

请用代码描述： 小王卖出一部价值6000的iPhone手机

### 操作步骤描述

1. 定义手机类
   1. 属性: 品牌(brand),价格(price)
   2. 要求:
      1. 提供带参构造,setXxx与getXxx方法
2. 定义人类(Person)
   1. 属性: 名称(name)
   2. 行为: 卖手机: Phone sellPhone() ,在方法内部使用匿名对象作为返回值
   3. 要求: 提供带参构造,setXxx和getXxx方法
3. 定义测试类Test
   1. 提供main方法
   2. 在main方法中
      1. 使用带参构造创建Person对象 p,name初始化为小王
      2. 调用对象p的sellPhone()方法,赋值给Phone phone变量
      3. 在main方法输出: 小王卖出一部价值6000的iPhone手机

**package** com.day10.test5;

**public** **class** Phone {

**private** String brand;

**private** String price;

**public** Phone() {

**super**();

}

**public** Phone(String brand, String price) {

**super**();

**this**.brand = brand;

**this**.price = price;

}

**public** String getBrand() {

**return** brand;

}

**public** **void** setBrand(String brand) {

**this**.brand = brand;

}

**public** String getPrice() {

**return** price;

}

**public** **void** setPrice(String price) {

**this**.price = price;

}

}

**package** com.day10.test5;

**public** **class** Person {

**private** String name;

**public** Phone sellPhone(String brand, String price) {

Phone phone = **new** Phone(brand, price);

**return** phone;

}

**public** Person() {

**super**();

}

**public** Person(String name) {

**super**();

**this**.name = name;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

}

**package** com.day10.test5;

**public** **class** Test {

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

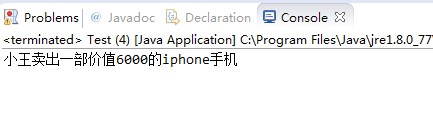
Person p = **new** Person("小王");

Phone phone = p.sellPhone("iphone", "6000");

System.***out***.println(p.getName() + "卖出一部价值" + phone.getPrice() + "的" + phone.getBrand() + "手机");

}

}



## 训练案例5

### 训练考核知识点

组合关系

### 训练描述

请用代码描述： 38岁的白百合有一部价值为8000元的iPhone的手机,她在使用者部手机给羽凡打电话

### 操作步骤描述

1. 定义手机类

* 1. 成员变量
     1. 品牌 brand
     2. 价格 price
  2. 成员方法
     1. 打电话 public void call(String name)
  3. 要求
     1. 提供无参和带参构造
     2. 提供setXxx和getXxx方法

1. 定义Person类
   1. 成员变量
      1. 姓名(name)
      2. 年龄(age)
      3. 手机(Phone)
   2. 成员方法: 打电话: public void callTo(Person p) 使用自己的手机给指定的人打电话
      1. 在方法内部调用手机的call(String name) 方法,给传入的人打电话
   3. 要求:
      1. 提供无参和带参构造
      2. 提供setXxx和getXxx方法
2. 定义测试类Test
   1. 提供main方法
   2. 在main方法中
      1. 使用带参构造创建手机对象 phone,品牌初始化为iPhone,价格初始化为8000
      2. 使用无参构造创建Person对象 yuFan,调用setName(String name)方法,把名称设置为羽凡
      3. 使用带参构造创建Person对象 baiBaiHe,名称初始化为白百合,年龄初始化为38,手机初始化,上面创建的phone对象
      4. baiBaiHe对象,调用call(Person p) 方法,传入yuFan对象
         1. 输出格式: 38岁的白百合使用8000元的iPhone的手机给羽凡打电话

**package** com.day10.test6;

**public** **class** Phone {

**private** String brand;

**private** String price;

**public** **void** call(String name) {

}

**public** Phone() {

**super**();

}

**public** Phone(String brand, String price) {

**super**();

**this**.brand = brand;

**this**.price = price;

}

**public** String getBrand() {

**return** brand;

}

**public** **void** setBrand(String brand) {

**this**.brand = brand;

}

**public** String getPrice() {

**return** price;

}

**public** **void** setPrice(String price) {

**this**.price = price;

}

}

**package** com.day10.test6;

**public** **class** Person {

**private** String name;

**private** String age;

**private** Phone phone;

**public** **void** callTo(Person p) {

**this**.phone.call(p.getName());

System.***out***.println(**this**.age + "岁的" + **this**.name + "正在使用价值" + **this**.phone.getPrice() + "元的" + **this**.phone.getBrand()

+ "手机给" + p.getName() + "打电话");

}

**public** Person() {

**super**();

}

**public** Person(String name, String age, Phone phone) {

**super**();

**this**.name = name;

**this**.age = age;

**this**.phone = phone;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getAge() {

**return** age;

}

**public** **void** setAge(String age) {

**this**.age = age;

}

**public** Phone getPhone() {

**return** phone;

}

**public** **void** setPhone(Phone phone) {

**this**.phone = phone;

}

}

**package** com.day10.test6;

**public** **class** Test {

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

Phone phone = **new** Phone("iPhone", "8000");

Person yuFan = **new** Person();

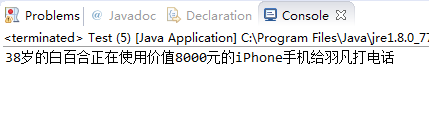
yuFan.setName("羽凡");

Person baiBaiHe = **new** Person("白百合", "38", phone);

baiBaiHe.callTo(yuFan);

}

}



## 训练案例6

### 训练考核知识点

继承的格式,子类访问父类的成员变量,子类调用父类的方法

### 训练描述

请用代码描述

棕色的Tom猫会吃和抓老鼠

白色的Jerry狗会吃和看家

要求: 把猫和狗的共性抽取到动物类中

### 操作步骤描述

1. 定义动物类(Animal)
   1. 成员变量
      1. 名称 name
      2. 颜色 color
   2. 成员方法
      1. 吃 eat()
         1. 打印格式: 白色的Tom动物在吃东西
   3. 要求:
      1. 成员变量不带权限修饰符,方法是公共权限的.
2. 定义猫类(Cat),继承Animal类
   1. 成员方法: 抓老鼠(catchMouse())
      1. 在方法中的输出格式为: 棕色的Tom猫在抓老鼠
3. 定义狗类(Dog),继承Animal类
   1. 成员方法: 看家 (lookHome())
      1. 在方法中的输出格式为: 白色的Jerry狗在看家
4. 定义测试类Test
   1. 提供main方法
   2. 在main方法中
      1. 使用空参构造,创建Cat对象c
      2. 把c对象的成员name赋值为Tom
      3. 把c对象的成员color赋值为 棕色
      4. 调用c对象成员方法eat方法
      5. 调用c对象的成员方法catchMouse()
      6. 使用空参构造,创建Dog对象 d
      7. 把d对象的成员name赋值为Jerry
      8. 把d对象的成员color赋值为 白色
      9. 调用d对象成员方法eat方法
      10. 调用d对象成员方法 lookHome方法

**package** com.day10.test7;

**public** **class** Animal {

String name;

String color;

**public** **void** eat() {

}

**public** Animal() {

**super**();

}

**public** Animal(String name, String color) {

**super**();

**this**.name = name;

**this**.color = color;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getColor() {

**return** color;

}

**public** **void** setColor(String color) {

**this**.color = color;

}

}

**package** com.day10.test7;

**public** **class** Cat **extends** Animal {

@Override

**public** **void** eat() {

System.***out***.println(color + "的" + name + "猫在吃东西");

}

**public** **void** catchMouse() {

System.***out***.println(color + "的" + name + "猫在抓老鼠");

}

**public** Cat() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** Cat(String name, String color) {

**super**(name, color);

// **TODO** Auto-generated constructor stub

}

}

**package** com.day10.test7;

**public** **class** Dog **extends** Animal {

**public** **void** eat() {

System.***out***.println(color + "的" + name + "狗在吃东西");

}

**public** **void** lookHome() {

System.***out***.println(color + "的" + name + "狗在看家");

}

**public** Dog() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** Dog(String name, String color) {

**super**(name, color);

// **TODO** Auto-generated constructor stub

}

}

**package** com.day10.test7;

**public** **class** Test {

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

Cat cat = **new** Cat();

cat.name = "Tom";

cat.color = "棕色";

cat.eat();

cat.catchMouse();

Dog dog = **new** Dog();

dog.name = "Jerry";

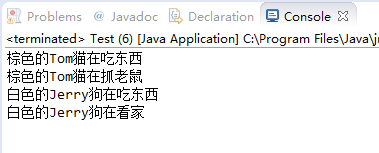
dog.color = "白色";

dog.eat();

dog.lookHome();

}

}



## 训练案例6

### 训练考核知识点

继承的格式,子类访问父类的成员变量,子类调用父类的方法

### 训练描述

请用代码描述

四个轮子的白色的汽车可以跑和加油

两个轮子的红色电动自行车可以跑和充电

要求: 把汽车和电动自行车的共性抽取到交通工具类中

### 操作步骤描述

1. 定义交通工具类(Vehicle)
   1. 成员变量
      1. 轮子个数 wheel
      2. 颜色 color
   2. 成员方法
      1. 跑 run()
         1. 打印格式: 4个轮子白色的车在跑
   3. 要求:
      1. 成员变量不带权限修饰符,方法是公共权限的.
2. 定义小汽车类(Car),继承Vehicle类
   1. 成员方法: 加油 (addGasoline() ))
      1. 在方法中的输出格式为: 4个轮子的白色小汽车在加油
3. 定义电动车类(ElectricBicycle),继承Vehicle类
   1. 成员方法: 充电 (charge())
      1. 在方法中的输出格式为: 2个轮子的黑色电动自行车在充电
4. 定义测试类Test
   1. 提供main方法
   2. 在main方法中
      1. 使用空参构造,创建Car 对象c
      2. 把c对象的成员wheel 赋值为 4
      3. 把c对象的成员color赋值为 白色
      4. 调用c对象成员方法run方法
      5. 调用c对象的成员方法addGasoline()
      6. 使用空参构造,创建ElectricBicycle对象 eb
      7. 把eb对象的成员wheel 赋值为2
      8. 把ed对象的成员color赋值为 黑色
      9. 调用ed对象成员方法run方法
      10. 调用ed对象成员方法 charge方法

**package** com.day10.test8;

**public** **class** Vehicle {

String wheel;

String color;

**public** **void** run() {

}

**public** Vehicle() {

**super**();

}

**public** Vehicle(String wheel, String color) {

**super**();

**this**.wheel = wheel;

**this**.color = color;

}

**public** String getWheel() {

**return** wheel;

}

**public** **void** setWheel(String wheel) {

**this**.wheel = wheel;

}

**public** String getColor() {

**return** color;

}

**public** **void** setColor(String color) {

**this**.color = color;

}

}

**package** com.day10.test8;

**public** **class** Car **extends** Vehicle {

**public** **void** addGasoline() {

System.***out***.println(wheel + "个轮子的" + color + "色小汽车在加油");

}

}

**package** com.day10.test8;

**public** **class** ElectricBicycle **extends** Vehicle {

**public** **void** charge(){

System.***out***.println(wheel + "个轮子的" + color + "色电动自行车在充电");

}

}

**package** com.day10.test8;

**public** **class** Test {

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

Car car = **new** Car();

car.wheel = "4";

car.color = "白色";

car.run();

car.addGasoline();

ElectricBicycle eBicycle = **new** ElectricBicycle();

eBicycle.wheel = "2";

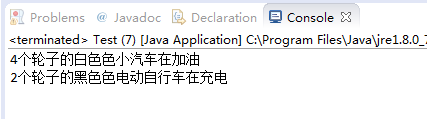
eBicycle.color = "黑色";

eBicycle.run();

eBicycle.charge();

}

}



# 关卡3

## 训练案例1

### 训练考核知识点

匿名对象调用方法，匿名对象作为实际参数,类名作为参数类型

### 训练描述

请用代码描述: 18岁的欧阳青青正在洗一件白色的LiNing牌的衣服

Person类：

**public** **void** washCloths(Cloth c){

System.***out***.println(**this**.age+"的"+**this**.name+"正在洗一件"+c.getColor()+"色的"+c.getBrand()+"牌衣服");

}

测试类:

**public** **class** Test08 {

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

Person person = **new** Person("欧阳青青",18);

person.washCloths(**new** Cloth("LiNing","白"));

}

}

## 训练案例2

### 训练考核知识点

组合关系

### 训练描述

请使用代码描述: 45岁的大男孩李晨有一个20岁的女友(范冰冰),李晨在逛街(和范冰冰)

女友类：

**public** **class** Girl {

**private** String name;

**private** **int** age;

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** Girl(String name, **int** age) {

**super**();

**this**.name = name;

**this**.age = age;

}

**public** Girl() {

**super**();

}

}

Person类：

**public** **void** shopping(){

System.***out***.println(age+"岁的大男孩"+name+"有一个"+girl.getAge()+"岁的女友("+girl.getName()

+"),"+name+"在逛街(和"+girl.getName()+")");

}

测试类：

**public** **class** Test09 {

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

Girl girl = **new** Girl("范冰冰",20);

Person person = **new** Person("李晨",45,girl);

person.shopping();

}

}

## 训练案例3

### 训练考核知识点

继承: 继承的格式,子类访问父类的成员变量,子类调用父类的方法

### 训练描述

请使用代码描述

18岁的服务员景甜,会说话,上菜

30岁的厨师秦俊杰,会说话,炒菜

要求: 把服务员类和厨师类的共性抽取到员工类中

Employee类：

**public** **class** Employee {

String name;

**int** age;

**public** Employee(String name, **int** age) {

**super**();

**this**.name = name;

**this**.age = age;

}

**public** **void** talk(){

System.***out***.println(age+"的"+name+"会说话");

}

}

Waiter类：

**public** **class** Waiter **extends** Employee {

**public** Waiter(String name, **int** age) {

**super**(name, age);

}

**public** **void** takeFood(){

System.***out***.println(age+"的服务员"+name+"会上菜");

}

}

Cook类：

**public** **class** Cook **extends** Employee {

**public** Cook(String name, **int** age) {

**super**(name, age);

}

**public** **void** makeDish(){

System.***out***.println(age+"的厨师"+name+"会炒菜");

}

}

测试类：

**public** **class** Test10 {

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

Waiter waiter = **new** Waiter("景甜",18);

Cook cook = **new** Cook("秦俊杰", 30);

waiter.talk();

waiter.takeFood();

cook.talk();

cook.makeDish();

}

}