**每日作业卷答案**

**就业班JavaSE第1天**

传智播客.黑马程序员

# 关卡1

## 训练案例1

### 训练描述

请阐述你对this关键字的理解

### 操作步骤答案

1. this一般出现在成员方法中

2. this代表本类的对象引用

3. 谁调用这方法,这个方法中this就是谁.

## 训练案例2

### 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

### 操作步骤答案

1. 请说出类名是 这类的对象 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,需要传入是 这个类的对象 .
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,该方法返回的值是这个类的对象 .

## 训练案例3

### 训练描述

请阐述你对匿名对象的理解

### 操作步骤答案

1. 匿名对象是指创建对象时，只有创建对象的语句，却没有把对象地址值赋值给某个变量。

2. 匿名对象没有变量名,只能使用一次,而有名对象有变量名,可以使用多次

3. 匿名对象可以作为方法实际参数; 也可以作为方法的返回值.

## 训练案例4

### 训练描述

请阐述你对组合关系的理解.

### 操作步骤答案

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，A类与B类叫做组合关系; 组合关系也被称之为has a(有一个)关系

2. 举例:

\* 人有一个笔记本电脑,此时人类和笔记本电脑类就是一种组合关系

\* 汽车有一个发动机, 此时汽车类和发动机类就是一种组合关系

## 训练案例5

### 训练描述

请阐述你对继承的理解

### 操作步骤答案

1. 在Java中，类的继承是指在一个现有类的基础上去构建一个新的类，构建出来的新类被称作子类，现有类被称作父类，子类会自动拥有父类所有可继承的属性和方法。

2. 当子类的成员变量和父类的成员变量不同名,且父类成员变量非私有的时候,子类可以直接访问成员变量; 如果父类成员变量私有了,子类就不能直接访问父类的成员变量了。

3. 在程序中通过对象调用方法时，会先在子类中查找有没有对应的方法，若子类中存在就会执行子类中的方法，若子类中不存在就会执行父类中相应的方法。

4. Java中继承的特点是:

1. Java中类是单继承的

2. Java支持多层继承

## 训练案例6

### 训练描述

请使用代码描述： 2岁的棕色加菲猫,在抓老鼠

### 操作步骤答案

/\*

1.定义Cat类

a)成员变量 名称(name), 颜色(color),年龄(age)

b)成员方法 抓老鼠(catchMouse())

c)提供空参和带参的构造方法

d)提供setter和getter方法

\*/

public class Cat **{**

//名称

private String name**;**

//颜色

private String color**;**

// 年龄

private int age**;**

// 成员方法

public void catchMouse**(){**

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

**}**

// 空参构造

public Cat**()** **{**

**}**

// 带参构造(全参构造)

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

**this.**name **=** name**;**

**this.**color **=** color**;**

**this.**age **=** age**;**

**}**

// setXxx与getXxx方法

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

**return** age**;**

**}**

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

**this.**age **=** age**;**

**}**

**}**

/\*

请使用代码描述： 2岁的棕色加菲猫,在抓老鼠

2. 创建测试类Test01

a)提供main方法

b)在main方法中

a)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

b)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

\*/

public class Test **{**

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

// 创建Cat对象

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

// 调用抓老鼠方法

c**.**catchMouse**();**

**}**

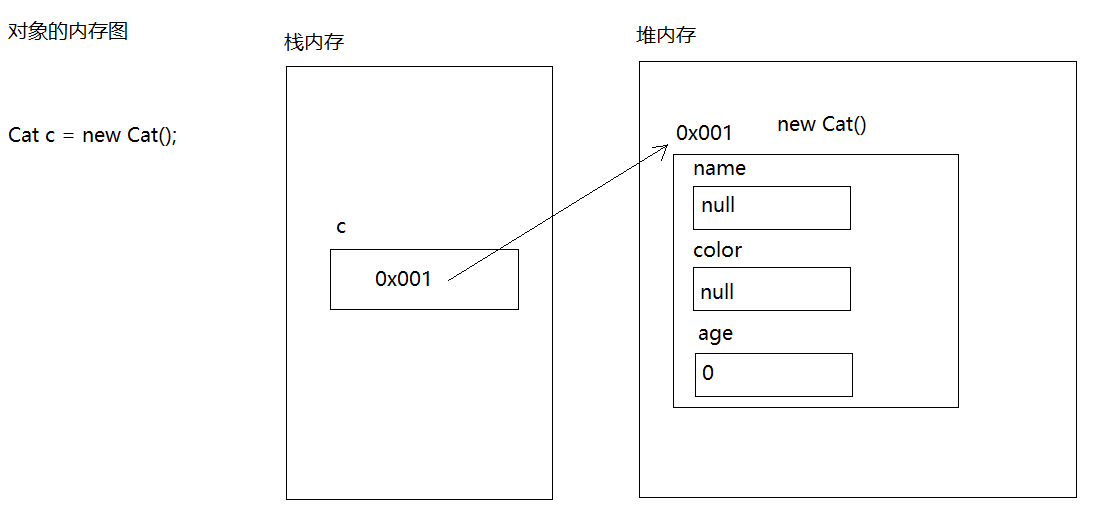
**}**

## 训练案例7

### 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1题的Cat类.

### 操作步骤答案



# 关卡2

## 训练案例1

### 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

### 操作步骤答案

/\*

1.定义手机类(Phone)

i.成员变量: 品牌(brand)

ii.成员方法: 玩游戏(void playGame())

iii.提供空参和带参构造方法

iv.提供setXxx和getXxx方法

\*/

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

// 品牌

**private** String brand;

// 成员方法

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

System.***out***.println("玩游戏");

}

// 空参构造

**public** Phone() {

}

// 有参构造

**public** Phone(String brand) {

**this**.brand = brand;

}

// setXxx与getXxx方法

**public** String getBrand() {

**return** brand;

}

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

**this**.brand = brand;

}

}

/\*

2.定义人类(Person)

i.成员变量: 姓名(name),年龄(age)

ii.成员方法: 使用手机玩游戏 (usePhone(Phone p))

iii.提供空参和带参构造方法

iv.提供setXxx和getXxx方法

\*/

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

// 姓名

**private** String name;

// 年龄

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

// 成员方法

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

System.***out***.print(age+"岁的"+name+"在使用"+p.getBrand()+"的手机");

p.playGame();

}

// 无参构造

**public** Person() {

}

// 带参构造

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

**this**.name = name;

**this**.age = age;

}

// setXxx与getXxx方法

**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;

}

}

/\*

请使用代码描述: 18岁的景甜在使用iPhone手机玩游戏

3.定义Test类

a)提供main方法

b)在main方法中

i.创建手机对象,并把手机品牌赋值为iPhone

ii.创建Person对象,并把名称赋值为景甜,年龄赋值为18

iii.调用Person对象的,usePhone方法,传入手机对象

iv.打印格式如下:

18岁的景甜在使用iPhone手机玩游戏

\*/

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

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

// i.创建手机对象,并把手机品牌赋值为iPhone

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

// ii.创建Person对象,并把名称赋值为景甜,年龄赋值为18

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

// iii.调用Person对象的,usePhone方法,传入手机对象

person.usePhone(phone);

}

}

## 训练案例2

### 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)

### 操作步骤答案

/\*

1.定义Person类

a)属性: 姓名(name)

b)行为: 唱歌(void sing(String song))

c)要求： 提供带参构造方法

\*/

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

// 姓名

**private** String name;

// 成员方法：唱歌

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

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

}

// 带参构造

**public** Person(String name) {

**this**.name = name;

}

}

/\*

请用代码描述: 周杰伦正在演唱菊花台歌曲(使用匿名对象)

2.定义测试类Test

a)提供main方法

b)在main方法中

i.使用带参构造创建Person对象,把name赋值为周杰伦

ii.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

\*/

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

// 提供main方法

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

// i.使用带参构造创建Person对象,把name赋值为周杰伦

// ii.使用上面创建的匿名对象调用唱歌的方法;

**new** Person("周杰伦").sing("菊花台");

}

}

## 训练案例3

### 训练描述

请用代码描述： 古力娜扎正在看择天记小说(使用匿名对象)

### 操作步骤答案

/\*

1.定义书类(Book)

a)属性: 名称(name)

b)要求: 提供带参构造,setXxx与getXxx方法

\*/

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

// 名称

**private** String name;

// 带参构造

**public** Book(String name) {

**this**.name = name;

}

// setXxx与getXxx方法

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

}

/\*

2.定义人类(Person)

a)属性: 名称(name)

b)行为: 看书: readBook(Book book)

c)要求: 提供带参构造

\*/

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

// 名称(name)

**private** String name;

// 看书: readBook(Book book)

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

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

}

// 带参构造

**public** Person(String name) {

**this**.name = name;

}

}

/\*

请用代码描述： 古力娜扎正在看择天记小说

3.定义测试类Test

a)提供main方法

b)在main方法中

i.使用带参构造创建Person对象,name初始化为古力娜扎

ii.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readBook(Book book)方法

c)输出格式

i.古力娜扎正在看择天记小说

\*/

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

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

// i.使用带参构造创建Person对象,name初始化为古力娜扎

// ii.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

**new** Person("古力娜扎").readBook(**new** Book("择天记"));

}

}

## 训练案例4

### 训练描述

请用代码描述： 小王卖出一部价值6000的iPhone手机

### 操作步骤答案

/\*

1.定义手机类

a)属性: 品牌(brand),价格(price)

b)要求:

i.提供带参构造,setXxx与getXxx方法

\*/

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

// 品牌

**private** String brand;

// 价格

**private** **int** price;

// 带参构造

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

**this**.brand = brand;

**this**.price = price;

}

// setXxx与getXxx方法

**public** String getBrand() {

**return** brand;

}

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

**this**.brand = brand;

}

**public** **int** getPrice() {

**return** price;

}

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

**this**.price = price;

}

}

/\*

2.定义人类(Person)

a)属性: 名称(name)

b)行为: 卖手机: Phone sellPhone() ,在方法内部使用匿名对象作为返回值

c)要求: 提供带参构造,setXxx和getXxx方法

\*/

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

// 名称

**private** String name;

// 卖手机: Phone sellPhone() ,在方法内部使用匿名对象作为返回值

**public** Phone sellPhone(){

**return** **new** Phone("iPhone", 6000);

}

// 带参构造

**public** Person(String name) {

**this**.name = name;

}

// setXxx和getXxx方法

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

}

/\*

请用代码描述： 小王卖出一部价值6000的iPhone手机

3.定义测试类Test

a)提供main方法

b)在main方法中

i.使用带参构造创建Person对象 p,name初始化为小王

ii.调用对象p的sellPhone()方法,赋值给Phone phone变量

iii.在main输出: 小王卖出一部价值6000的iPhone手机

\*/

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

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

// i.使用带参构造创建Person对象 p,name初始化为小王

Person p = **new** Person("小王");

// ii.调用对象p的sellPhone()方法,赋值给Phone phone变量

Phone phone = p.sellPhone();

// iii.输出: 小王卖出一部价值6000的iPhone手机

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

### 操作步骤答案

/\*

1. 定义手机类

a)成员变量

i.品牌 brand

ii.价格 price

b)成员方法

i.打电话 public void call(String name)

c)要求

i.提供无参和带参构造

ii.提供setXxx和getXxx方法

\*/

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

// 品牌 brand

**private** String brand;

// 价格 price

**private** **int** price;

// 打电话 public void call(String name)

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

System.***out***.println("给"+name+"打电话");

}

// 无参和带参构造

**public** Phone() {

}

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

**this**.brand = brand;

**this**.price = price;

}

// setXxx和getXxx方法

**public** String getBrand() {

**return** brand;

}

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

**this**.brand = brand;

}

**public** **int** getPrice() {

**return** price;

}

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

**this**.price = price;

}

}

/\*

2.定义Person类

a)成员变量

i.姓名(name)

ii.年龄(age)

iii.手机(Phone)

b)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

i.在方法内部调用手机的call(String name) 方法,给传入的人打电话

c)要求:

i.提供无参和带参构造

ii.提供setXxx和getXxx方法

\*/

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

// a)成员变量

// i.姓名(name)

**private** String name;

// ii.年龄(age)

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

// iii.手机(Phone)

**private** Phone phone;

// b)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

// i.在方法内部调用手机的call(String name) 方法,给传入的人打电话

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

System.***out***.print(age+"岁的"+name+"使用"+phone.getPrice()+"元的"+phone.getBrand()+"的手机部手机");

phone.call(p.name);

}

// c)要求:

// i.提供无参和带参构造

**public** Person() {

}

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

**this**.name = name;

**this**.age = age;

**this**.phone = phone;

}

// ii.提供setXxx和getXxx方法

**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** Phone getPhone() {

**return** phone;

}

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

**this**.phone = phone;

}

}

/\*

请使用代码描述： 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

3.定义测试类Test

a)提供main方法

b)在main方法中

i.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

ii.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

iii.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

iv.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

\*/

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

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

// i.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

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

// ii.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

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

yuFan.setName("羽凡");

// iii.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

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

// iv.baiBaiHe对象,调用call(Person p) 方法,传入yuFan对象

baiBaiHe.callTo(yuFan);

// 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

}

}

## 训练案例6

### 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

### 操作步骤答案

/\*

1.定义动物类(Animal)

a)成员变量

i.名称 name

ii.颜色 color

b)成员方法

i.吃 eat()

打印格式: 白色的Tom动物在吃东西

c)要求:

i.成员变量不带权限修饰符,方法是公共权限的.

\*/

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

// a)成员变量

// i.名称 name

String name;

// ii.颜色 color

String color;

//b)成员方法

// i.吃 eat()

// 打印格式: 白色的Tom动物在吃东西

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

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

}

}

/\*

2.定义猫类(Cat),继承Animal类

a)成员方法: 抓老鼠(catchMouse())

i.在方法中的输出格式为: 棕色的Tom猫在抓老鼠

\*/

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

// 抓老鼠(catchMouse())

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

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

}

}

/\*

3.定义狗类(Dog),继承Animal类

a)成员方法: 看家 (lookHome())

i.在方法中的输出格式为: 白色的Jerry狗在看家

\*/

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

// 看家 (lookHome())

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

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

}

}

/\*

4.定义测试类Test

a)提供main方法

b)在main方法中

i.使用空参构造,创建Cat对象c

ii.把c对象的成员name赋值为Tom

iii.把c对象的成员color赋值为 棕色

iv.调用c对象成员方法eat方法

v.调用c对象的成员方法catchMouse()

vi.使用空参构造,创建Dog对象 d

vii.把d对象的成员name赋值为Jerry

viii.把d对象的成员color赋值为 白色

ix.调用d对象成员方法eat方法

x.调用d对象成员方法 lookHome方法

\*/

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

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

// i.使用空参构造,创建Cat对象c

Cat c = **new** Cat();

// ii.把c对象的成员name赋值为Tom

c.name = "Tom";

// iii.把c对象的成员color赋值为 棕色

c.color = "棕色";

// iv.调用c对象成员方法eat方法

c.eat();

// v.调用c对象的成员方法catchMouse()

c.catchMouse();

// vi.使用空参构造,创建Dog对象 d

Dog d = **new** Dog();

// vii.把d对象的成员name赋值为Jerry

d.name = "Jerry";

// viii.把d对象的成员color赋值为 白色

d.color = "白色";

// ix.调用d对象成员方法eat方法

d.eat();

// x.调用d对象成员方法 lookHome方法

d.lookHome();

}

}

## 训练案例7

### 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

### 操作步骤答案

/\*

1.定义交通工具类(Vehicle)

a)成员变量

i.轮子个数 wheel

ii.颜色 color

b)成员方法

i.跑 run()

1.打印格式: 4个轮子白色的车在跑

c)要求:

i.成员变量不带权限修饰符,方法是公共权限的.

\*/

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

// a)成员变量

// i.轮子个数 wheel

**int** wheel;

// ii.颜色 color

String color;

//b)成员方法

// i.跑 run()

// 1.打印格式: 四个轮子白色的车在跑

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

System.***out***.println(wheel+"个轮子"+color+"的车在跑");

}

}

/\*

2.定义小汽车类(Car),继承Vehicle类

a)成员方法: 加油 (addGasoline() ))

i.在方法中的输出格式为: 4个轮子的白色小汽车在加油

\*/

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

// a)成员方法: 加油 (addGasoline() ))

// i.在方法中的输出格式为: 4个轮子的白色小汽车在加油

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

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

}

}

/\*

3.定义电动车类(ElectricBicycle),继承Vehicle类

a)成员方法: 充电 (charge())

i.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

\*/

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

// a)成员方法: 充电 (charge())

// i.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

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

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

}

}

/\*

4.定义测试类Test

c)提供main方法

d)在main方法中

i.使用空参构造,创建Car 对象c

ii.把c对象的成员wheel 赋值为 4

iii.把c对象的成员color赋值为 白色

iv.调用c对象成员方法run方法

v.调用c对象的成员方法addGasoline()

vi.使用空参构造,创建ElectricBicycle对象 eb

vii.把eb对象的成员wheel 赋值为2

viii.把ed对象的成员color赋值为 黑色

ix.调用ed对象成员方法run方法

x.调用ed对象成员方法 charge方法

\*/

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

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

// i.使用空参构造,创建Car 对象c

Car c = **new** Car();

// ii.把c对象的成员wheel 赋值为 4

c.wheel = 4;

// iii.把c对象的成员color赋值为 白色

c.color = "白色";

// iv.调用c对象成员方法run方法

c.run();

// v.调用c对象的成员方法addGasoline()

c.addGasoline();

// vi.使用空参构造,创建ElectricBicycle对象 eb

ElectricBicycle eb = **new** ElectricBicycle();

// vii.把eb对象的成员wheel 赋值为2

eb.wheel = 2;

// viii.把ed对象的成员color赋值为 黑色

eb.color = "黑色";

// ix.调用ed对象成员方法run方法

eb.run();

// x.调用ed对象成员方法 charge方法

eb.charge();

}

}

# 关卡3

## 训练案例1

### 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

### 操作步骤

1. 定义衣服（Clothes）类

1. 属性： 颜色（color），品牌(brand)

2. 提供带参构造和setXxx和getXxx方法

2. 定义人（Person）类

1. 属性： 名称（name）,年龄（age）

2. 行为： 洗衣服,把衣服对象作为参数传入

输出格式为： 18岁的欧阳青青正在洗一件白色的LiNing牌的衣服

3. 提供：带参构造方法

3. 定义测试类Test

1. 提供main方法

2. 在main方法中

1. 使用人类的带参构造创建人匿名对象，把name赋值为欧阳青请，把age赋值为18

2. 使用人的匿名对象调用洗衣服的方法，传出使用衣服类的带参构造创建的衣服的匿名对象，把衣服对象的颜色赋值为白色，品牌赋值为LiNing

### 操作步骤答案

/\*

1. 定义衣服（Clothes）类

1. 属性： 颜色（color），品牌(brand)

2. 提供带参构造和setXxx和getXxx方法

\*/

**public** **class** Clothes {

// 颜色（color）

**private** String color;

// 品牌(brand)

**private** String brand;

// 带参构造

**public** Clothes(String color, String brand) {

**this**.color = color;

**this**.brand = brand;

}

// setXxx和getXxx方法

**public** String getColor() {

**return** color;

}

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

**this**.color = color;

}

**public** String getBrand() {

**return** brand;

}

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

**this**.brand = brand;

}

}

/\*

2. 定义人（Person）类

1. 属性： 名称（name）,年龄（age）

2. 行为： 洗衣服,把衣服对象作为参数传入

输出格式为： 18岁的欧阳青青正在洗一件白色的LiNing牌的衣服

3. 提供：带参构造方法

\*/

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

// 名称

**private** String name;

// 年龄

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

// 洗衣服

**public** **void** wash(Clothes c){

System.***out***.println(age+"岁的"+name+"正在洗一件"+c.getColor()+"的"+c.getBrand()+"牌的衣服");

}

// 带参构造

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

**this**.name = name;

**this**.age = age;

}

}

/\*

3. 定义测试类Test

1. 提供main方法

2. 在main方法中

1. 使用人类的带参构造创建人匿名对象，把name赋值为欧阳青请，把age赋值为18

2. 使用人的匿名对象调用洗衣服的方法，传出使用衣服类的带参构造创建的衣服的匿名对象，把衣服对象的颜色赋值为白色，品牌赋值为LiNing

\*/

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

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

// 1. 使用人类的带参构造创建人匿名对象，把name赋值为欧阳青请，把age赋值为18

// 2. 使用人的匿名对象调用洗衣服的方法，传出使用衣服类的带参构造创建的衣服的匿名对象，把衣服对象的颜色赋值为白色，品牌赋值为LiNing

**new** Person("欧阳青请", 18).wash(**new** Clothes("白色", "LiNing"));

}

}

## 训练案例2

### 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)

### 操作步骤

1. 定义Girl类

1. 属性： 名称：name,年龄 age

2. 提供带参构造和setXxx和getXxx方法

2. 定义Boy类

1. 属性： 名称：name,年龄：age，女友（girlFriend）

2. 行为： public void walking()

输出格式为： 45岁的大男孩李晨在和20岁的小女友范冰冰逛街

3. 提供满参构造和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. 使用满参构造创建Gilr对象 girl,把name初始化为范冰冰，年龄初始化为20

2. 使用满参构造创建Boy 对象 boy,把name初始化为李晨，年龄初始化为45，女友初始化为girl

3. 调用boy的walking()方法

### 操作步骤答案

/\*

1. 定义Girl类

1. 属性： 名称：name,年龄 age

2. 提供带参构造和setXxx和getXxx方法

\*/

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

// 名称

**private** String name;

// 年龄

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

// 带参构造

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

**this**.name = name;

**this**.age = age;

}

// setXxx和getXxx方法

**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;

}

}

/\*

2. 定义Boy类

1. 属性： 名称：name,年龄：age，女友（girlFriend）

2. 行为： public void walking()

输出格式为： 45岁的大男孩李晨在和20岁的小女友范冰冰逛街

3. 提供满参构造和setXxx和getXxx方法

\*/

**public** **class** Boy {

// 名称

**private** String name;

// 年龄

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

// 女友

**private** Girl girlFriend;

// 逛街

**public** **void** walking(){

System.***out***.println(age+"岁的大男孩"+name+"在和"+girlFriend.getAge()+"岁的小女友"+girlFriend.getName()+"逛街 ");

}

// 满参构造

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

**this**.name = name;

**this**.age = age;

**this**.girlFriend = girlFriend;

}

// setters和getters

**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 getGirlFriend() {

**return** girlFriend;

}

**public** **void** setGirlFriend(Girl girlFriend) {

**this**.girlFriend = girlFriend;

}

}

/\*

3. 定义测试类Test

1. 提供main方法

2. 在main方法中

1. 使用满参构造创建Gilr对象 girl,把name初始化为范冰冰，年龄初始化为20

2. 使用满参构造创建Boy 对象 boy,把name初始化为李晨，年龄初始化为45，女友初始化为girl

3. 调用boy的walking()方法

\*/

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

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

// 1. 使用满参构造创建Gilr对象 girl,把name初始化为范冰冰，年龄初始化为20

Girl girl = **new** Girl("范冰冰", 20);

// 2. 使用满参构造创建Boy 对象 boy,把name初始化为李晨，年龄初始化为45，女友初始化为girl

Boy boy = **new** Boy("李晨", 45, girl);

// 3. 调用boy的walking()方法

boy.walking();

}

}

## 训练案例3

### 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

### 操作步骤描述

1. 定义Employee类

1. 属性： 名称：name,年龄 age

2. 行为： 说话（say）

输出格式： 18岁的员工景甜在说话

3. 要求： 成员变量不加权限private

2. 定义Waiter类,继承Employee类

1. 行为： 上菜 public void serving()

输出格式为： 18岁的服务员景甜正在上菜

3. 定义Chef类,继承Employee类

1. 行为： 炒菜 public void cooking()

输出格式为： 30岁的厨师秦俊杰正在炒菜

4. 定义测试类Test

1. 提供main方法

2. 在main方法中

1. 使用Waiter的空参构造创建对象 w

2. 给对象w的name赋值为景甜

3. 给对象w的age赋值为18

4. 调用对象w的说话方法

5. 调用对象w的上菜方法

6. 使用Chef的空参构造创建对象 c

7. 给对象c的name赋值为秦俊杰

8. 给对象c的age赋值为30

9. 调用对象c的说话方法

10. 调用对象c的炒菜方法

### 操作步骤答案

/\*

1. 定义Employee类

1. 属性： 名称：name,年龄 age

2. 行为： 说话（say）

输出格式： 18岁的员工景甜在说话

3. 要求： 成员变量不加权限private

\*/

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

// 名称：name

String name;

// 年龄 age

**int** age;

// 说话（say）

**public** **void** say(){

System.***out***.println(age+"岁的员工"+name+"在说话");

}

}

/\*

2. 定义Waiter类,继承Employee类

1. 行为： 上菜 public void serving()

输出格式为： 18岁的服务员景甜正在上菜

\*/

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

// 1. 行为： 上菜 public void serve()

**public** **void** serving(){

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

}

}

/\*

3. 定义Chef类,继承Employee类

1. 行为： 炒菜 public void cooking()

输出格式为： 30岁的厨师秦俊杰正在炒菜

\*/

**public** **class** Chef **extends** Employee{

// 行为： 炒菜 public void cooking()

**public** **void** cooking() {

System.***out***.println(age+"岁的厨师"+name+"正在炒菜");

}

}

/\*

4. 定义测试类Test

1. 提供main方法

2. 在main方法中

1. 使用Waiter的空参构造创建对象 w

2. 给对象w的name赋值为景甜

3. 给对象w的age赋值为18

4. 调用对象w的说话方法

5. 调用对象w的上菜方法

6. 使用Chef的空参构造创建对象 c

7. 给对象c的name赋值为秦俊杰

8. 给对象c的age赋值为30

9. 调用对象c的说话方法

10. 调用对象c的炒菜方法

\*/

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

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

// 1. 使用Waiter的空参构造创建对象 w

Waiter w = **new** Waiter();

// 2. 给对象w的name赋值为景甜

w.name = "景甜";

// 3. 给对象w的age赋值为18

w.age = 18;

// 4. 调用对象w的说话方法

w.say();

// 5. 调用对象w的上菜方法

w.serving();

// 6. 使用Chef的空参构造创建对象 c

Chef c = **new** Chef();

// 7. 给对象c的name赋值为秦俊杰

c.name = "秦俊杰";

// 8. 给对象c的age赋值为30

c.age = 18;

// 9. 调用对象c的说话方法

c.say();

// 10. 调用对象c的炒菜方法

c.cooking();

}

}