操作题

> **操作题，无需表现在作业答案中，自己琢磨和练习即可。**（如有留下痕迹的必要，我会具体给出。）

1. 读程序题：读程序，然后分析过程和结果，提供必要的文字说明。

```
public class Test {
    public static void main(String[] args) {
        Sub sub = new Sub();
    }
}
class Base{
    static {
        System.out.println("base 静态代码块");
    }
    {
        System.out.println("base 构造代码块");
    }
    public Base(){
        System.out.println("base无参构造");
    }
}
class Sub extends Base{
    static {
        System.out.println("sub 静态代码块");
    }
    {
        System.out.println("sub 构造代码块");
    }
    public Sub(){
        System.out.println("sub 无参构造");
    }
}
```

（1）子类加载会按照从上到下的顺序加载父类和子类，于是创建sub的时候会先进行父类类加载再到子类的类加载：

```
    static {
        System.out.println("base 静态代码块");
    }
     static {
        System.out.println("sub 静态代码块");
    }
```

（2）接着调用子类构造器也会按照从上到下的顺序调用父类和子类的构造器：

```
    {
        System.out.println("base 构造代码块");
    }
    public Base(){
        System.out.println("base无参构造");
    }
    {
        System.out.println("sub 构造代码块");
    }
    public Sub(){
        System.out.println("sub 无参构造");
    }
```

于是结果为：

```
    base 静态代码块
    sub 静态代码块
    base 构造代码块
    base无参构造
    sub 构造代码块
    sub 无参构造
```

由运行结果可知正确：

```
PS D:\code\homework>  & 'C:\eve\.jdks\jdk-11.0.14\bin\java.exe' '-cp' 'C:\
cd6434e4eeb4685d8d59e9fb1a27eee\redhat.java\jdt_ws\homework_f6d44879\bin'
base 静态代码块
sub 静态代码块
base 构造代码块
base无参构造
sub 构造代码块
sub 无参构造
PS D:\code\homework>
```

2. 读程序题：读程序，然后分析过程和结果，提供必要的文字说明。

```
public class Obj3 extends Obj1 {
    Obj2 ob2 = new Obj2();
    public Obj3(){

        System.out.println("obj3");
    }
    public static void main(String[] args) {
        Obj3 obj3 = new Obj3();
    }
}
class Obj1 {
    Obj2 ob2 = new Obj2();
    public Obj1() {
        System.out.println("Obj1");
```

```
    }
}
class Obj2 {
    public Obj2() {
        System.out.println("obj2");
    }
}
```

```
Obj3 obj3 = new Obj3();
    (1)Obj2 ob2 = new Obj2();
            (1.1)public Obj2() {
                    System.out.println("obj2");
                }
    (2)public Obj1() {
         System.out.println("Obj1");
       }
    (3)Obj2 ob2 = new Obj2();
            (3.1)public Obj2() {
                    System.out.println("obj2");
                }
    (4)public Obj3(){
          System.out.println("obj3");
       }
所以：
obj2
Obj1
obj2
obj3
```

```
PS D:\code\homework>  d:; cd 'd:\code\homework'; & 'C:\eve\.jdks\jdk-11.0.14\bin\java.exe'
ode\User\workspaceStorage\8cd6434e4eeb4685d8d59e9fb1a27eee\redhat.java\jdt_ws\homework_f6d
Obj3'
obj2
Obj1
obj2
obj3
PS D:\code\homework>
```

简答题

以下简答题，直接将答案写在题目下面即可。（都是一些概念，虽然我们学得是技术，但基本的概念还是需要记忆的）

1. 继承使用什么关键字？继承中的两个类是什么关系？

```
（1）extends
（2）父子关系
```

2. Java中一个类能否**直接**继承两个类？Java中一个类就只能有一个父类吗？

```
（1）不能多继承
（2）不是可以通过父类继承其他的类来让子类继承更多的
```

3. 如果一个类没有明确继承某个类，那么它就没有父类吗？

```
No，Object是所有类的祖先类
```

4. A继承B，C也继承B，那么A和C**（血缘上是兄弟姐妹关系）**这两个类**从继承上**来说有什么关系？

```
毫无关系
```

5. 引用数据类型能否发生数据类型转换？如果能发生，前提是什么？

```
（1）自动：父类引用指向子类对象
（2）强制：子类引用指向父类对象需要强制类型转换
```

6. 从继承方向的角度上，描述引用数据类型的自动类型转换和强制类型转换，并指出它们的特殊叫法。

```
（1）自动：向下      父类名 引用名 = new 子类名();
（2）强制：向上      子类名 引用名 = (子类名)new 父类名();
```

7. 描述protected访问权限级别（从同包，不同包的角度回答）'

```
不同包下的子类可以访问
同包更能
```

8. 私有成员可以被子类继承吗？

```
可以，但是访问受限
```

9. 构造器能够被子类继承吗？

```
不能
```

10. 静态成员可以被子类继承吗？

```
不能，是共享
```

11. 子类对象隐式初始化的条件是什么？

```

```

12. 描述一下**对象名点访问成员变量**的访问机制
    - 访问范围
    - 访问结果

    从上述两个角度来说明这个机制。

```
（1） 访问范围
    - 父类引用，指向父类对象
        范围：父类
    - 父类引用，指向子类对象
        范围：父类
    - 子类引用，指向子类对象
        范围：父类+子类
（2） 访问结果
    - 父类引用，指向父类对象
        范围：父类
    - 父类引用，指向子类对象
        范围：父类
    - 子类引用，指向子类对象
        范围：父类+子类
```

编程题

继承基础语法练习

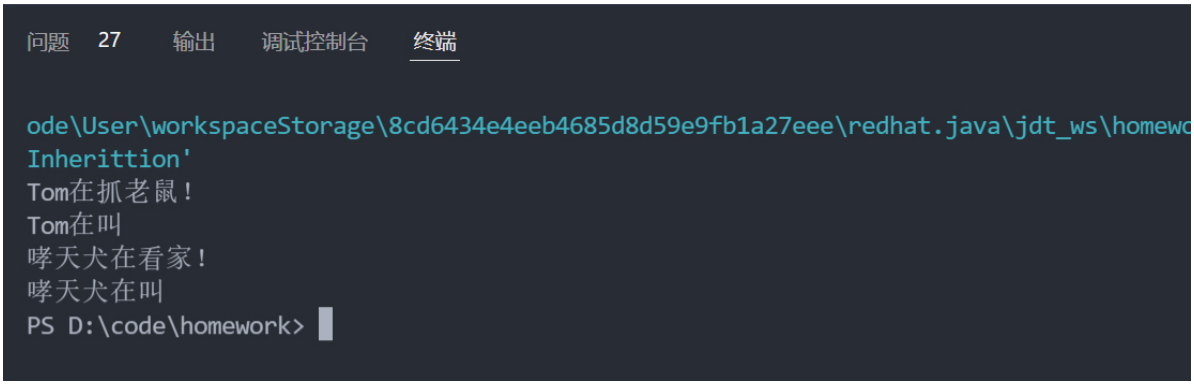
> 明确记住，继承中的两个类应该有is-a关系

提供以下两个动物需要描述，请用你的知识来编写代码
猫：姓名，年龄，颜色，可以叫，可以抓老鼠
狗：姓名，年龄，性别，可以叫，可以看门
分析这个案例，设计出合适的继承体系。

最后思考：人类研究出来了机器人，它们也有姓名，年龄，颜色等属性，可以套用本题中的继承体系吗？

```
package com.cskaoyan.homework.day08;

/**
     提供以下两个动物需要描述，请用你的知识来编写代码
         猫：姓名，年龄，颜色，可以叫，可以抓老鼠
         狗：姓名，年龄，性别，可以叫，可以看门
    分析这个案例，设计出合适的继承体系。
    最后思考：人类研究出来了机器人，
             它们也有姓名，年龄，颜色等属性，可以套用本题中的继承体系吗？
*/

public class Inherittion {
    public static void main(String[] args) {
        Cat cat = new Cat("Tom",3,"Black");
        Dog dog = new Dog("哮天犬",1000,"Black");
        cat.catchMouse();
        cat.bark();
        dog.guardHouse();
        dog.bark();
    }


}

// Pat类 ：  姓名，年龄，可以叫
class Pat{
    String name;
    int age;
    String color;
```

```
    public Pat(String name,int age,String color){
        this.name = name;
        this.age = age;
        this.color=color;
    }

    public void bark(){
        System.out.println(this.name+"在叫");
    }
}

// 猫：姓名，年龄，颜色，可以叫，可以抓老鼠
class Cat extends Pat{
    // String name;
    // int age;
    // String color;

    // public void bark(){
    //     System.out.println(this.name+"在叫");
    // }
    public Cat(String name,int age,String color){
        super(name, age, color);
    }


    public void catchMouse(){
        System.out.println(this.name+"在抓老鼠！");
    }
}

// 狗：姓名，年龄，性别，可以叫，可以看门
class Dog extends Pat{
    // String name;
    // int age;
    // String color;

    // public void bark(){
    //     System.out.println(this.name+"在叫");
    // }

    public Dog(String name,int age,String color){
        super(name, age, color);
    }

    public void  guardHouse(){
        System.out.println(this.name+"在看家！");
    }
}
```



机器人也可以继承宠物的。它也有名字、年龄（即当前时间-生产日期）、颜色；也会叫；最主要的机器人也可以当宠物符合面对对象的思维

完成GUI-阶段练习一

详见文档说明

```
//UserDao类

    //name存在返回true
    public boolean isNameExist(User user) {
        for (int i = 0; i < this.users.length && this.users[i] !=
null; i++) {
            if
(user.getUsername().equals(this.users[i].getUsername())) {
                return true;
            }
        }
        return false;
    }

    //name对应密码正确返回true
    public boolean isPasswordTrue(User user) {
        int index = -1;
        for (int i = 0; i < this.users.length && this.users[i] !=
null; i++) {
            if
(user.getUsername().equals(this.users[i].getUsername())) {
                index = i;
            }
        }
        if (index >= 0 &&
user.getPassword().equals(this.users[index].getPassword())) {
            return true;
        }
```

```
            return false;
        }
```

```
//UserController类

    public int judgeLogin(User user) {
        if (!userDao.isNameExist(user)) {
            return 1;
        } else if (!userDao.isPasswordTrue(user)) {
            return 2;
        }
        return 0;
    }
```


王道学生信息管理系统
王道学生信息管理系统
用户名:
新冉冉
密 码:
消息提示
用户名不存在
确定
登陆
退出
输入
新冉冉
1231


王道学生信息管理系统
欢迎光临学生管理系统
查询学生
新增学生
保存数据至文件
退出程序
王道训练营