**实验4继承**

**一、实验时间**

2023年10月10日星期二

**二、实验目的**

1、进一步理解Java面向对象程序设计的基本思想；

2、理解继承的概念、作用与意义；

3、掌握访问控制符在继承中的应用；

4、理解什么是方法的覆盖，以及方法的覆盖与方法的重载的区别；

5、掌握Java继承关系中构造方法的应用。

**三、实验内容**

1、实现一个Animal类，该类包含type、name、color。

定义该类的三个构造方法：无参构造方法、带type和name参数的构造方法、带type、name、color参数的构造方法；

定义该类的每个属性的get和set方法；

定义一个抽象方法sleep方法；

定义一个抽象方法方法wakeUp方法；

定义一个抽象方法walk；

2、实现一个Cat类，该类继承Animal类。

该类中新增属性age，定义该类的两个构造方法，无参构造方法、带参数type、name、color、age属性的构造方法（调用父类的构造方法初始化从父类继承的type、name、color属性）。1

分别覆盖父类中的sleep方法、wakeUp方法，其功能分别打印输出“The cat is sleeping!”、“The cat wakes up!”；1

实现toString方法，返回Cat类的相关信息（包含从父类继承得到数据属性信息）；1

实现Animal类中的walk方法，打印输出“The cat is walking silently.”。1

3、实现一个Horse类，该类继承Animal类。

该类中新增属性height，定义该类的两个构造方法，无参构造方法、带参数type、name、color、height属性的构造方法（调用父类的构造方法初始化从父类继承的type、name、color属性）。

分别覆盖父类中的sleep方法、wakeUp方法，其功能分别打印输出“The horse is sleeping!”、“The horse wakes up!”；

实现toString方法，返回Horse类的相关信息（包含从父类继承得到数据属性信息）；

实现Animal类中的walk方法，打印输出“The horse is walking and shouting.”。

4、定义一个测试类AnimalTest，定义一个ArrayList对象成员animalList；在该类的main方法中分别定义3个猫对象和3个马对象，并添加到animalList；通过循环依次打印所有猫对象和所有马对象的信息，并分别调用这些对象的walk方法和wakeup方法。

5、在第4步创建猫对象和马对象时，初始化参数要求通过键盘输入。

**四、实验过程**

1、实验原理分析（写出你采用的实验理论基础）（10分）

本次实验主要利用了抽象方法和抽象类，如何进行重写和如何进行对函数的重载。如何继承父类，调用子类对象。

2、实验设计（写出实验的算法设计）（25分）

3、实验过程（写出实验过程中遇到的问题及解决的思路和方法）（30分）

４、实验源码（20分）

//Animal.java

package 源代码存储.Java.实验4.CODE;

abstract class Animal

{

    String type;

    String name;

    String color;

    //无参构造函数方法

    public Animal()

    {

        type="动物";

        name="xxx";

        color="x色";

    }

    //构造函数重载1;

    public Animal(String ty,String name)

    {

        type=ty;

        this.name=name;

    }

    //构造函数重载2

    public Animal(String ty,String name,String co)

    {

        type=ty;

        this.name=name;

        color=co;

    }

    //get函数

    String getType()

    {

        return type;

    }

    String getName()

    {

        return name;

    }

    String getColor()

    {

        return color;

    }

    //设置函数

    void setType(String ty)

    {

        type=ty;

    }

    void setName(String nm)

    {

        name=nm;

    }

    void setColor(String co)

    {

        color=co;

    }

    //抽象方法

    abstract void sleep();

    abstract void wakeUp();

    abstract void walk();

}

class Cat extends Animal

{

    int age;//设置年龄

    public Cat()

    {

        type="猫";

        name="xxx";

        color="x色";

        age=0;

    }

    //重载设置方法

    public Cat(String ty,String nm,String co,int age)

    {

        type=ty;

        name=nm;

        color=co;

        this.age=age;

    }

    void sleep()

    {

        System.out.println("The cat is sleeping!");

    }

    void wakeUp()

    {

        System.out.println("The cat wakes up!");

    }

    void walk()

    {

        System.out.println("The cat is walking silently.");

    }

    public String toString()

    {

        return "这是一只"+age+"岁"+color+"色"+"叫做"+name+"的"+type;

    }

}

class Horse extends Animal

{

    int height;//设置年龄

    public Horse()

    {

        type="马";

        name="xxx";

        color="x色";

        height=0;

    }

    //重载设置方法

    public Horse(String ty,String nm,String co,int height)

    {

        type=ty;

        name=nm;

        color=co;

        this.height=height;

    }

    void sleep()

    {

        System.out.println("The horse is sleeping!");

    }

    void wakeUp()

    {

        System.out.println("The horse wakes up!");

    }

    void walk()

    {

        System.out.println("The horse is walking and shouting..");

    }

    public String toString()

    {

        return "这是一只"+height+"高"+color+"色"+"叫做"+name+"的"+type;

    }

}

//AnimalTest.java

package 源代码存储.Java.实验4.CODE;

import java.util.Scanner;

public class AnimalTest {

    public static void main(String[] args) {

        Animal animallist[]=new Animal[6];

        Scanner sc=new Scanner(System.in);

        String str1;

        String str2;

        String str3;

        int xx;

        //猫

        for(int i=0;i<3;i++)

        {

            System.out.println("请输入第"+(i+1)+"只猫的信息（顺序为：type name color age）");

            str1=sc.next();

            str2=sc.next();

            str3=sc.next();

            xx=sc.nextInt();

            animallist[i]=new Cat(str1,str2,str3,xx);

        }

        //马

        for(int i=0;i<3;i++)

        {

            System.out.println("请输入第"+(i+1)+"只马的信息（顺序为：type name color height）");

            str1=sc.next();

            str2=sc.next();

            str3=sc.next();

            xx=sc.nextInt();

            animallist[3+i]=new Horse(str1,str2,str3,xx);

        }

        for(int i=0;i<6;i++)

        {

            System.out.println(animallist[i]);

            animallist[i].walk();

            animallist[i].wakeUp();

        }

        sc.close();

    }

}

５、运行结果（要求对结果分析，分别对输入出错和输入正确的结果进行分析）（10分）

请输入第1只猫的信息（顺序为：type name color age）

BC NMA Black 23 (输入)

请输入第2只猫的信息（顺序为：type name color age）

BC NMb White 24 (输入)

请输入第3只猫的信息（顺序为：type name color age）

BC NMD WHite 24 (输入)

请输入第1只马的信息（顺序为：type name color height）

Qianlima NDW WHITE 44 (输入)

请输入第2只马的信息（顺序为：type name color height）

SQW DWL WLL 99 (输入)

请输入第3只马的信息（顺序为：type name color height）

SWE DSC WAL 20 (输入)

这是一只23岁Black色叫做NMA的BC

The cat is walking silently.

The cat wakes up!

这是一只24岁White色叫做NMb的BC

The cat is walking silently.

The cat wakes up!

这是一只24岁WHite色叫做NMD的BC

The cat is walking silently.

The cat wakes up!

这是一只44高WHITE色叫做NDW的Qianlima

The horse is walking and shouting..

The horse wakes up!

这是一只99高WLL色叫做DWL的SQW

The horse is walking and shouting..

The horse wakes up!

这是一只20高WAL色叫做DSC的SWE

The horse is walking and shouting..

The horse wakes up!

6、实验总结（根据实验的过程和结果分析写出你对实验原理的理解）（5分）

通过本次实验，我们进一步理解了Java面向对象程序设计的基本思想。然而，实验也存在一些不足之处。例如，在实现方法覆盖时，未能充分理解方法覆盖与方法重载的区别，导致在编程过程中出现了一些误解。

评分表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 项目 | 总分 | 评分标准 | 实际得分 |
| 1 | 实验原理分析 | 10 | 对实验全过程涉及的知识点应用进行有效分析 |  |
| 2 | 实验设计 | 25 | 类的封装机制、继承机制、多态机制、异常机制的设计（10分），类的属性成员的设计（5分），类的方法成员的设计（5分），测试部分输入的设计（3分）、测试部分输出的设计（2分） |  |
| 3 | 实验过程 | 30 | 对遇到的问题进行分析（10分），解决思路（5分）、方案（5分）、过程（10分）， |  |
| 4 | 实验源码 | 20 | 实验源码的书写规范性（2分）、标识符的命名规范（2分）、是否有注释（2分）；代码的正确性（14分） |  |
| 5 | 运行结果 | 10 | 是否有运行截图（5分），是否对结果进行分析（5分） |  |
| 6 | 实验总结 | 5 | 对实验是否进行了有效总结，主要体现在对知识点的理解，实验过程的总结、知识点应用总结 |  |
| 总分 |  | | | |
| 实验评价 |  | | | |