

1、课程名称：实例分析：宠物商店



实例要求

- 实现一个宠物商店，在宠物商店中可以有多种（由用户决定数量）宠物，试表示出此种关系，并要求可以根据宠物的关键字查找找到相应的宠物信息。所需要的宠物信息自行设计。

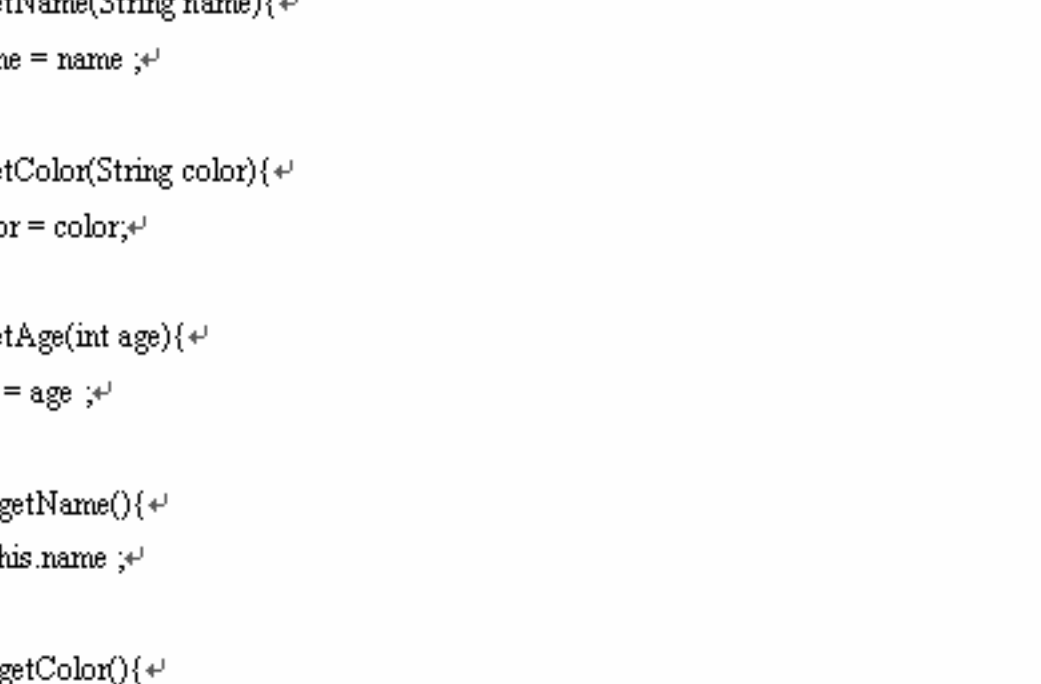
3、本实例主要采用的知识

- 接口
- 对象数组

4、具体内容

分析

- 1、本要求中提示宠物的信息可以自行设计，所以此时简单设计出三个属性：名字、颜色、年龄。
- 2、宠物的类别很多，例如：猫、狗等都属于宠物，所以宠物应该是一个标准。
- 3、在宠物商店中，只要是符合了此宠物标准的就都应该可以放进宠物商店之中。
- 4、宠物商店中要保存多种宠物，则肯定应该是一个宠物的对象数组，宠物的个数由用户决定的话，则应该在创建宠物商店的时候，就已经分配好宠物的个数。



```
E-MAIL: mldnqa@163.com www.MLDNJAVA.cn

interface Pet{ // 定义宠物接口
    public String getName();
    public String getColor();
    public int getAge();
}

class Cat implements Pet{ // 猫是宠物，实现接口
    private String name; // 宠物名字
    private String color; // 宠物颜色
    private int age; // 宠物年龄
    public Cat(String name,String color,int age){
        this.setName(name);
        this.setColor(color);
        this.setAge(age);
    }

    public void setName(String name){
        this.name = name;
    }

    public void setColor(String color){
        this.color = color;
    }

    public void setAge(int age){
        this.age = age;
    }

    public String getName(){
        return this.name;
    }

    public String getColor(){
        return this.color;
    }

    public int getAge(){
        return this.age;
    }
}

class Dog implements Pet{ // 狗是宠物，实现接口
    private String name; // 宠物名字
    private String color; // 宠物颜色
    private int age; // 宠物年龄
    public Dog(String name,String color,int age){
        this.setName(name);
        this.setColor(color);
        this.setAge(age);
    }

    public void setName(String name){
        this.name = name;
    }

    public void setColor(String color){
        this.color = color;
    }

    public void setAge(int age){
        this.age = age;
    }

    public String getName(){
        return this.name;
    }

    public String getColor(){
        return this.color;
    }

    public int getAge(){
        return this.age;
    }
}

class PetShop{ // 宠物商店
    private Pet[] pets; // 保存一组宠物
    private int foot;

    public PetShop(int len){
        if(len>0){
            this.pets = new Pet[len]; // 开辟数组大小
        }else{
            this.pets = new Pet[1]; // 至少开辟一个空间
        }
    }

    public boolean add(Pet pet){ // 增加的是一个宠物
        if(this.foot<this.pets.length){
            this.pets[this.foot] = pet; // 增加宠物
            this.foot++;
            return true;
        }else{
            return false;
        }
    }

    public Pet[] search(String keyWord){
        // 应该确定有多少个宠物符合要求
        Pet p[] = null;
        int count = 0; // 记录下会有多少个宠物符合查询结果
        for(int i=0;i<this.pets.length;i++){
            if(this.pets[i]!=null){ // 表示此位置有宠物
                if(this.pets[i].getName().indexOf(keyWord)!=-1){
                    if(this.pets[i].getColor().indexOf(keyWord)!=-1){
                        p[count++] = this.pets[i];
                    }
                }
            }
        }
        return p;
    }
}

public class PetShopDemo{
    public static void main(String args[]){
        PetShop ps = new PetShop(5); // 五个宠物
        ps.add(new Cat("白猫","白色的",2)); // 增加宠物，成功
        ps.add(new Cat("黑猫","黑色的",3)); // 增加宠物，成功
        ps.add(new Cat("花猫","花色的",3)); // 增加宠物，成功
        ps.add(new Dog("拉布拉多","黄色的",3)); // 增加宠物，成功
        ps.add(new Dog("金毛","金色的",2)); // 增加宠物，成功
        ps.add(new Dog("黄狗","黑色的",2)); // 增加宠物，失败
        print(ps.search("黑"));

        public static void print(Pet p[]){
            for(int i=0;i<p.length;i++){
                if(p[i]!=null){
                    System.out.println(p[i].getName() + " , " + p[i].getColor()
                        + " , " + p[i].getAge());
                }
            }
        }
    }
}
```

```
E-MAIL: mldnqa@163.com www.MLDNJAVA.cn

    public void setName(String name){
        this.name = name;
    }

    public void setColor(String color){
        this.color = color;
    }

    public void setAge(int age){
        this.age = age;
    }

    public String getName(){
        return this.name;
    }

    public String getColor(){
        return this.color;
    }

    public int getAge(){
        return this.age;
    }
}

class Dog implements Pet{ // 狗是宠物，实现接口
    private String name; // 宠物名字
    private String color; // 宠物颜色
    private int age; // 宠物年龄
    public Dog(String name,String color,int age){
        this.setName(name);
        this.setColor(color);
        this.setAge(age);
    }

    public void setName(String name){
        this.name = name;
    }

    public void setColor(String color){
        this.color = color;
    }

    public void setAge(int age){
        this.age = age;
    }

    public String getName(){
        return this.name;
    }

    public String getColor(){
        return this.color;
    }

    public int getAge(){
        return this.age;
    }
}

class PetShop{ // 宠物商店
    private Pet[] pets; // 保存一组宠物
    private int foot;

    public PetShop(int len){
        if(len>0){
            this.pets = new Pet[len]; // 开辟数组大小
        }else{
            this.pets = new Pet[1]; // 至少开辟一个空间
        }
    }

    public boolean add(Pet pet){ // 增加的是一个宠物
        if(this.foot<this.pets.length){
            this.pets[this.foot] = pet; // 增加宠物
            this.foot++;
            return true;
        }else{
            return false;
        }
    }

    public Pet[] search(String keyWord){
        // 应该确定有多少个宠物符合要求
        Pet p[] = null;
        int count = 0; // 记录下会有多少个宠物符合查询结果
        for(int i=0;i<this.pets.length;i++){
            if(this.pets[i]!=null){ // 表示此位置有宠物
                if(this.pets[i].getName().indexOf(keyWord)!=-1){
                    if(this.pets[i].getColor().indexOf(keyWord)!=-1){
                        p[count++] = this.pets[i];
                    }
                }
            }
        }
        return p;
    }
}

public class PetShopDemo{
    public static void main(String args[]){
        PetShop ps = new PetShop(5); // 五个宠物
        ps.add(new Cat("白猫","白色的",2)); // 增加宠物，成功
        ps.add(new Cat("黑猫","黑色的",3)); // 增加宠物，成功
        ps.add(new Cat("花猫","花色的",3)); // 增加宠物，成功
        ps.add(new Dog("拉布拉多","黄色的",3)); // 增加宠物，成功
        ps.add(new Dog("金毛","金色的",2)); // 增加宠物，成功
        ps.add(new Dog("黄狗","黑色的",2)); // 增加宠物，失败
        print(ps.search("黑"));

        public static void print(Pet p[]){
            for(int i=0;i<p.length;i++){
                if(p[i]!=null){
                    System.out.println(p[i].getName() + " , " + p[i].getColor()
                        + " , " + p[i].getAge());
                }
            }
        }
    }
}
```

```
E-MAIL: mldnqa@163.com www.MLDNJAVA.cn

    public void setName(String name){
        this.name = name;
    }

    public void setColor(String color){
        this.color = color;
    }

    public void setAge(int age){
        this.age = age;
    }

    public String getName(){
        return this.name;
    }

    public String getColor(){
        return this.color;
    }

    public int getAge(){
        return this.age;
    }
}

class Dog implements Pet{ // 狗是宠物，实现接口
    private String name; // 宠物名字
    private String color; // 宠物颜色
    private int age; // 宠物年龄
    public Dog(String name,String color,int age){
        this.setName(name);
        this.setColor(color);
        this.setAge(age);
    }

    public void setName(String name){
        this.name = name;
    }

    public void setColor(String color){
        this.color = color;
    }

    public void setAge(int age){
        this.age = age;
    }

    public String getName(){
        return this.name;
    }

    public String getColor(){
        return this.color;
    }

    public int getAge(){
        return this.age;
    }
}

class PetShop{ // 宠物商店
    private Pet[] pets; // 保存一组宠物
    private int foot;

    public PetShop(int len){
        if(len>0){
            this.pets = new Pet[len]; // 开辟数组大小
        }else{
            this.pets = new Pet[1]; // 至少开辟一个空间
        }
    }

    public boolean add(Pet pet){ // 增加的是一个宠物
        if(this.foot<this.pets.length){
            this.pets[this.foot] = pet; // 增加宠物
            this.foot++;
            return true;
        }else{
            return false;
        }
    }

    public Pet[] search(String keyWord){
        // 应该确定有多少个宠物符合要求
        Pet p[] = null;
        int count = 0; // 记录下会有多少个宠物符合查询结果
        for(int i=0;i<this.pets.length;i++){
            if(this.pets[i]!=null){ // 表示此位置有宠物
                if(this.pets[i].getName().indexOf(keyWord)!=-1){
                    if(this.pets[i].getColor().indexOf(keyWord)!=-1){
                        p[count++] = this.pets[i];
                    }
                }
            }
        }
        return p;
    }
}

public class PetShopDemo{
    public static void main(String args[]){
        PetShop ps = new PetShop(5); // 五个宠物
        ps.add(new Cat("白猫","白色的",2)); // 增加宠物，成功
        ps.add(new Cat("黑猫","黑色的",3)); // 增加宠物，成功
        ps.add(new Cat("花猫","花色的",3)); // 增加宠物，成功
        ps.add(new Dog("拉布拉多","黄色的",3)); // 增加宠物，成功
        ps.add(new Dog("金毛","金色的",2)); // 增加宠物，成功
        ps.add(new Dog("黄狗","黑色的",2)); // 增加宠物，失败
        print(ps.search("黑"));

        public static void print(Pet p[]){
            for(int i=0;i<p.length;i++){
                if(p[i]!=null){
                    System.out.println(p[i].getName() + " , " + p[i].getColor()
                        + " , " + p[i].getAge());
                }
            }
        }
    }
}
```

类图



5、总结

在本程序中实际上最重要的就是接口的设计，只要接口设计的足够合理，则程序开发会有很高的灵活性，使用接口可以进行解耦操作。