

# 中北大学软件学院

## 实 验 报 告

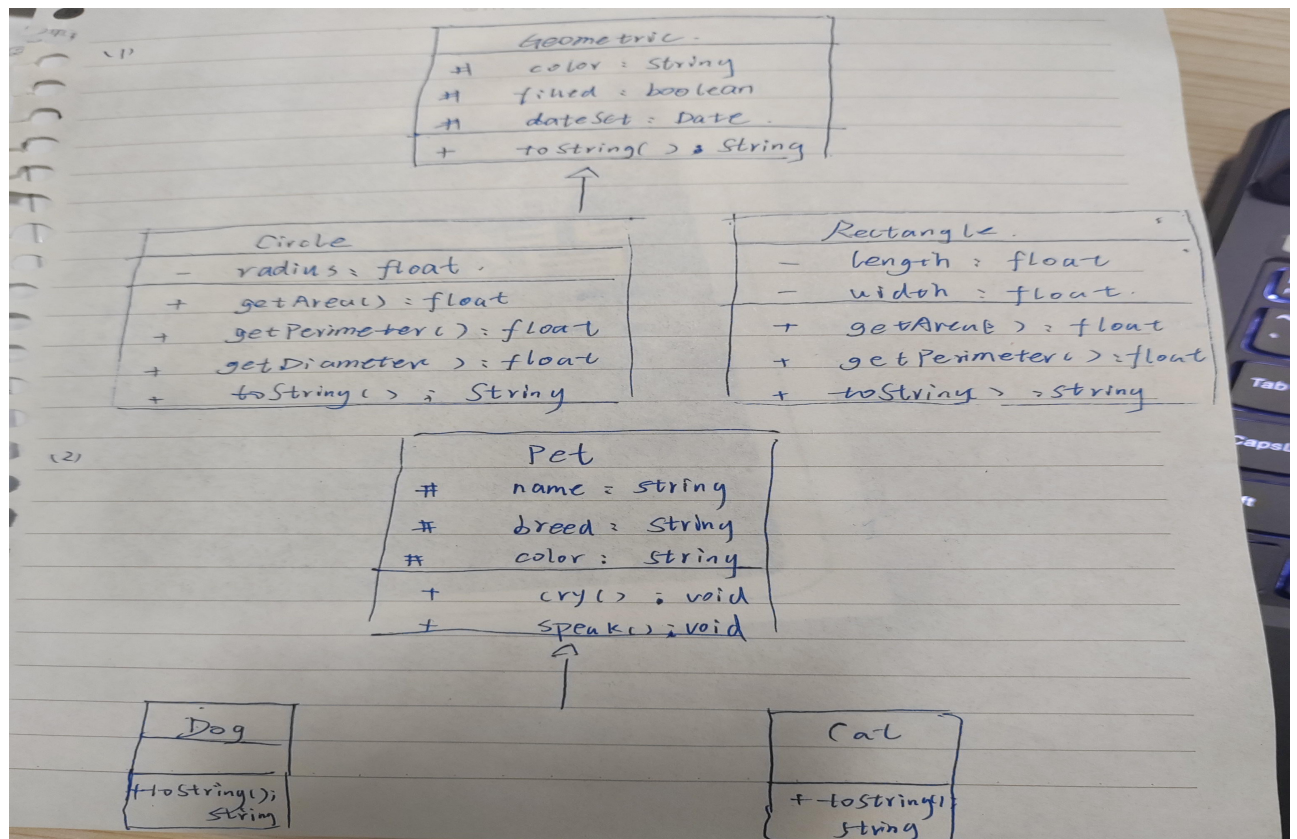
专 业:	软件工程
课程名称:	面向对象程序设计
班 级:	22130403
学 号:	2213040340
姓 名:	张恒瑜
辅导教师:	李华玲

2021 年 9 月 制

成绩：

实验时间	2023 年 10 月 8 日 14 时至 18 时	学时数
1. 实验名称		
继承与多态		
2. 实验目的		
(1) 熟练掌握继承的实现机制和语法；		
(2) 掌握子类对象实例化与构造器；		
(3) 理解对象的上转型机制和多态的含义，学会在实际应用中使用多态；		
3. 训练知识点集群		
(1) 继承（子类对象实例化与构造器、super、覆盖等）		
(2) 多态（上转型对象、多态的含义、多态的实现）		
4. 实验内容		
内容 1：（必做）		
(1) 定义一个几何图形类 Geometric，该类中包含私有化成员属性：图形颜色 color、图形是否填充 filled、图形用 Date 类获取系统时间），并且重写 toString 方法，输出形状、是否填充及创建时间		
(2) 定义一个子类圆类 Circle，该类中包含私有化成员属性：radius，包含获取圆面积的方法 getArea（），获取圆周长（），获取圆直径的方法 getDiameter（），重写 toString（）方法		
(3) 定义一个子类长方形类 Rectangle，该类中包含私有化成员属性：长方形的长 length，宽 width，包含获取长（），获取长方形周长的方法 getPerimeter（），重写 toString（）方法		
(4) 给出测试类		
内容 2：利用继承和多态的知识完成宠物商店宠物的“自我介绍”程序。		
要求：可以输出许多宠物的信息。输出格式为：我是一只狗。我叫乐乐，是一只黑色的牧羊犬。我会汪汪汪…我是一只猫。我叫咪咪，是一只白色的波斯猫。我会喵喵喵…。		
内容 3：（学生自由创造）		
根据你理解的多态的含义，自己写一个有关多态的案例，并测试运行。请在下面描述你的案例：		

## 5. 根据问题抽象出类图



## 6. 实验源代码

( 1 )



Circle.java



Geometric.java



Rectangle.java



TestShape.java

```
package nuc_zhy0340_shape;
```

```
import java.util.Date;
```

```
public class Geometric {  
    protected String color;  
    protected boolean filled;  
    protected Date dataSet = new Date();
```

```
    public Geometric() {  
        super();  
    }
```

```
    public Geometric(String color, boolean filled) {  
        super();  
        this.color = color;  
        this.filled = filled;  
    }
```

```
    public Geometric(String color, boolean filled, Date dataSet) {  
        super();  
        this.color = color;  
        this.filled = filled;  
        this.dataSet = dataSet;  
    }
```

```
    public String getColor() {  
        return color;  
    }
```

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

```
    public boolean isFilled() {
        return filled;
    }

    public void setFilled(boolean filled) {
        this.filled = filled;
    }

    public Date getDateSet() {
        return dateSet;
    }

    public void setDateSet(Date dateSet) {
        this.dateSet = dateSet;
    }

    public String toString() {
        return "Geometric [color=" + color + ", filled=" + filled + "];"
    }
}
```

---

```
package nuc_zhy0340_shape;

public class Circle extends Geometric{
    private float radius;

    public Circle (float radius, String color, boolean filled) {
        super(color, filled);
        this.radius = radius;
    }

    public float getRadius() {
        return radius;
    }

    public void setRadius(float radius) {
        this.radius = radius;
    }

    public float getArea() {
        return 3.14f * radius * radius;
    }
}
```

```

    }

    public float getPerimeter () {
        return 2 * 3.14f * radius;
    }

    public float getDiameter () {
        return 2 * radius;
    }

    public String toString() {
        return "Circle [radius=" + radius + ", color=" + color + ", filled=" + filled
是： " + getDateSet()

        + ",圆的面积是： " + getArea() + ", 圆的周长是： " + getPerime

径是： " + getDiameter()

        + "]";
    }

}

```

---

```

package nuc_zhy0340_shape;

public class Rectangle extends Geometric{
    private float length;
    private float width;

    public Rectangle (float length, float width, String color, boolean filled) {
        super(color, filled);
        this.length = length;
        this.width = width;
    }

    public float getLength() {
        return length;
    }

    public void setLength(float length) {
        this.length = length;
    }
}

```

```

    }

    public float getWidth() {
        return width;
    }

    public void setWidth(float width) {
        this.width = width;
    }

    public float getArea () {
        return length*width;
    }

    public float getPerimeter () {
        return 2 * (length + width);
    }

    public String toString() {
        return "Rectangle [length=" + length + ", width=" + width + ", color=" +
+ filled
        + ", 创建时间是: " + getDateSet() + ", 长方形的面积是: " + ge
形的周长是: " + getPerimeter() + "];"
    }
}

```

---

```

package nuc_zhy0340_shape;

import java.util.Scanner;

public class TestShape {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        while (true) {
            System.out.println("*****");

```

```

System.out.println("*****1.圆*****");

System.out.println("*****2.长方形*****");

System.out.println("*****3.结束*****");
System.out.println("*****");
System.out.println("请输入你要进行的选择：");

Scanner sc = new Scanner(System.in);
int choose = sc.nextInt();
switch (choose) {
    case 1 : {
        System.out.println("请输入圆的半径，颜色以及是否填满：");

        float radius = sc.nextFloat();
        String color = sc.next();
        boolean filled = sc.nextBoolean();
        Geometric circle = new Circle(radius, color ,filled);
        System.out.println(circle.toString());
        break;
    }
    case 2 : {
        System.out.println("请输入长方形的长，宽，颜色以及是否填满");

        float length = sc.nextFloat();
        float width = sc.nextFloat();
        String color = sc.next();
        boolean filled = sc.nextBoolean();
        Geometric rectangle = new Rectangle(length, width, color ,filled);
        System.out.println(rectangle.toString());
        break;
    }
    case 3 : {
        return;
    }
}
}
}
}
}

```



( 2 )



Pet.java



Dog.java



Cat.java



TestAnimals.java

```
package nuc_zhy0340_animals;
```

```
public class Pet {
    protected String name;
    protected String breed;
    protected String color;

    public Pet() {
        super();
    }

    public Pet(String name, String breed, String color) {
        this.name = name;
        this.breed = breed;
        this.color = color;
    }

    public void cry() {}

    public void speak () {
        System.out.print("我是一只" + toString() + "。我叫" + name + "，是一
+ breed + " 。" + "我会");
        cry();
    }
}
```

```
package nuc_zhy0340_animals;
```

```
public class Dog extends Pet{
    public Dog (String name, String breed, String color) {
        super(name, breed, color);
    }

    public void cry() {
        System.out.println("汪汪汪...");
    }
}
```

```
    public String toString() {  
        return "狗";  
    }  
}
```

---

```
package nuc_zhy0340_animals;  
  
public class Cat extends Pet{  
    public Cat (String name, String breed, String color) {  
        super(name, breed, color);  
    }  
  
    public void cry () {  
        System.out.println("喵喵喵...");  
    }  
  
    public String toString () {  
        return "猫";  
    }  
}
```

---

```
package nuc_zhy0340_animals;  
  
import java.util.Scanner;  
  
public class TestAnimals {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Scanner sc = new Scanner(System.in);  
        while (true) {  
            System.out.println("*****");  
            System.out.println("****选择你的宠物*****");  
  
            System.out.println("*****1.狗子*****");  
  
            System.out.println("*****2.猫儿*****");  
        }  
    }  
}
```

```

        System.out.println("*****3.退出*****");
        System.out.println("*****");
        int choose = sc.nextInt();
        switch (choose) {
            case 1 : {
                System.out.println("请输入修勾的名字，品种，颜色：");
                String name = sc.next();
                String breed = sc.next();
                String color = sc.next();
                Pet dog = new Dog(name, breed, color);
                dog.toString();
                dog.speak();
                break;
            }
            case 2 : {
                System.out.println("请输入修猫的名字，品种，颜色：");
                String name = sc.next();
                String breed = sc.next();
                String color = sc.next();
                Pet cat = new Cat(name, breed, color);
                cat.toString();
                cat.speak();
                break;
            }
            case 3 : {}
                return;
        }
    }
}
}

```

（3）这里是自己做的多态代码

很简单，就是一种名字相同的方法在父类和子类中拥有不同的效果  
多态所做的就是在继承的关系下，通过父类的引用调用在子类中重写的方法



Polymorphism.java

```
package nuc_zhy0340_Poly;

public class Polymorphism {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        //                关于多态                //

        //1.必须在继承体系下，子类必须继承父类

        //2.子类必须要对父类中方法进行重写

        //3.通过父类的引用调用重写的方法

        Animal cat = new Cat();
        Animal dog = new Dog();
        Animal ani = new Animal();

        ani.sound();
        dog.sound();
        cat.sound();
    }
}

//这里是原始代码

class Animal {
    public void sound() {
        System.out.println("The animal makes a sound");
    }
}

class Cat extends Animal {
    public void sound() {
        System.out.println("The cat meows");
    }
}

class Dog extends Animal {
    public void sound() {
        System.out.println("The dog barks");
    }
}
```

```
}  
}
```

## 7.实验运行结果截图及心得体会

( 1 )

```
*****2.长方形*****  
*****3.结束*****  
*****  
请输入你要进行的选择:  
1  
请输入圆的半径, 颜色以及是否填满:  
13.14  
purple  
true  
Circle [radius=13.14, color=purple, filled=true, 创建时间是: Sun Oct 08 17:31:20 CST 2023, 圆的面积是: 542.1512, 圆  
*****  
*****1.圆*****  
*****2.长方形*****  
*****3.结束*****  
*****  
请输入你要进行的选择:  
2  
请输入长方形的长, 宽, 颜色以及是否填满:  
33  
22  
pink  
false  
Rectangle [length=33.0, width=22.0, color=pink, filled=false, 创建时间是: Sun Oct 08 17:31:33 CST 2023, 长方形的  
*****  
*****1.圆*****  
*****2.长方形*****  
*****3.结束*****  
*****  
请输入你要进行的选择:  
3
```

( 2 )

```

*****选择你的宠物*****
*****1.狗子*****
*****2.猫儿*****
*****3.退出*****
*****
1
请输入修勾的名字，品种，颜色：
旺财
柴犬
黄黑色
我是一只狗。我叫旺财，是一只黄黑色的柴犬 。我会汪汪汪...
*****
*****选择你的宠物*****
*****1.狗子*****
*****2.猫儿*****
*****3.退出*****
*****
2
请输入修猫的名字，品种，颜色：
小神
橘猫
金黄色
我是一只猫。我叫小神，是一只金黄色的橘猫 。我会喵喵喵...
*****
*****选择你的宠物*****
*****1.狗子*****
*****2.猫儿*****
*****3.退出*****
*****

```

```
The animal makes a sound  
The dog barks  
The cat meows
```

## 心得体会：

做完这个实验对继承和多态有了自己的理解，清楚的了解父类和子类之间的关系，再接再厉！





