

Java06-封装继承与多态

所有代码存放于sixth文件夹当中

Task1

Java不支持多继承:

如果两个父类有同名方法或属性，子类继承时会产生冲突，不易管理。

可以用多实现来解决多继承的复杂问题。

Task2

代码如下(不再单开一个文件):

```
// 定义一个图形接口
interface Shape {
    double getPerimeter(); // 求周长
    double getArea();      // 求面积
}

// 圆形类
class Circle implements Shape {
    double r;
    public Circle(double r) { this.r = r; }
    @Override
    public double getPerimeter() { return 2 * Math.PI * r; }
    @Override
    public double getArea() { return Math.PI * r * r; }
}

// 矩形类
class Rectangle implements Shape {
    double w, h;
    public Rectangle(double w, double h) { this.w = w; this.h = h; }
    @Override
    public double getPerimeter() { return 2 * (w + h); }
    @Override
    public double getArea() { return w * h; }
}

// 三角形类
class Triangle implements Shape {
    double a, b, c;
    public Triangle(double a, double b, double c) { this.a = a; this.b = b; this.c = c; }
    @Override
    public double getPerimeter() { return a + b + c; }
    @Override
    public double getArea() {
        double p = getPerimeter() / 2;
        return Math.sqrt(p * (p - a) * (p - b) * (p - c)); // 海伦公式
    }
}
```

```
}  
}
```

Task3

public (公有)：被修饰的成员可以被任何其他类访问。例：public int age; 任何地方都能访问。

private (私有)：被修饰的成员只能在当前类内部访问，外部类无法访问。例：private int age; 只能在本类中使用，外部不能直接访问。

```
public class BankAccount {  
    // TODO 修改属性的可见性  
    private String accountNumber;  
    private String accountHolder;  
    private double balance;  
    private String password; // 敏感信息, 需要严格保护  
  
    BankAccount(String accountNumber, String accountHolder, double initialBalance,  
String password) {  
        //TODO  
    }  
  
    void deposit(double amount) {  
        //TODO  
    }  
  
    boolean withdraw(double amount, String inputPassword) {  
        //TODO  
    }  
  
    boolean transfer(BankAccount recipient, double amount, String inputPassword) {  
        //TODO  
    }  
  
    double getBalance() {  
        //TODO  
    }  
  
    String getAccountInfo() {  
        //TODO  
    }  
    // 只需修改可见性  
    public boolean validatePassword(String inputPassword) {  
        return true;  
    }  
    // 只需修改可见性  
    public boolean validateAmount(double amount) {  
        return true;  
    }  
}
```

