java06

Task1

关于java为什么不支持多继承:

假如两个父类有方法名，形参均相同，而方法体不同的方法，子类在调用时就不知道使用哪个了~~~

Task2

这个是常量类

```
package task6;

public class constant {
    public static final double PI = 3.14;
}
```

这个是接口

```
package task6;

public interface demo {
    double getLength(double a, double b);
    double getArea(double a, double b);
}
```

以下两个是实现类

```
package task6;

public class JuXing implements demo{
    @Override
    public double getLength(double a, double b) {
        return 2 * (a + b);
    }

    @Override
    public double getArea(double a, double b) {
        return a * b;
    }
    public static void main(String[] args) {
        JuXing yuan = new JuXing();
        double a = 5;
        double b = 5;
        System.out.print("矩形的宽是:");
        System.out.println(a);
```

```
        System.out.print("矩形的高是:");
        System.out.println(b);
        System.out.print("矩形的周长是:");
        System.out.println(yuan.getLength(a, b));
        System.out.print("矩形的面积是:");
        System.out.println(yuan.getArea(a, b));
    }
}
```

```
package task6;

public class Yuan implements demo{
    //对几何图形进行建模并完成圆，三角形，矩形的周长，面积计算
    //其实还有另一种做法，使用接口来完成，请你给出使用接口的解答
    @Override
    public double getLength(double r, double i) {
        return 2 * constant.PI * r;
    }

    @Override
    public double getArea(double r, double i) {
        return constant.PI * r * r;
    }
    public static void main(String[] args) {
        Yuan yuan = new Yuan();
        int r = 5;
        System.out.print("圆的周长是:");
        System.out.println(yuan.getLength(r, 0));
        System.out.print("圆的面积是:");
        System.out.println(yuan.getArea(r, 0));
    }
```

P.S. 感觉圆，矩形，三角形需要的形参不一样（圆只需要半径，矩形需要长宽，三角需要三边）所以我只能做出这个简化的,大佬轻喷~~~ **QWQ**

Task3

权限修饰符：

private：只能本类。

缺省：同一个包。

protected：同一个包 + 子孙类

public：任意位置

(黑马老师说一般就使用private和public就好)

以下是银行账户的代码

```
package task6;
```

```
    public class BankAccount {
        private String accountNumber;
        private String accountHolder;
        private double balance;
        private String password;

        BankAccount(String accountNumber, String accountHolder, double
initialBalance, String password) {
            System.out.println("创建账户成功！");
            this.accountNumber = accountNumber;
            this.accountHolder = accountHolder;
            this.balance = initialBalance;
            this.password = password;
            System.out.println(this.getAccountInfo());
        }

        void deposit(double amount) {
            balance = amount + balance;
            System.out.println("存款成功!当前金额为" + balance + "元");
        }

        boolean withdraw(double amount, String inputPassword) {
            if(inputPassword.equals(password)){
                if(amount <= balance){
                    balance = balance - amount;
                    System.out.println("取款成功！当前余额为" + balance + "元");
                    return true;
                }else{
                    System.out.println("取款失败！余额不足");
                    return false;
                }
            }else{
                System.out.println("取款失败！密码错误");
                return false;
            }
        }

        boolean transfer(BankAccount recipient, double amount, String
inputPassword) {
            if (inputPassword.equals(password)){
                if (amount <= balance){
                    balance = balance - amount;
                    recipient.balance = recipient.balance + amount;
                    System.out.println("转账成功！当前余额为" + balance + "元");
                    return true;
                }else{
                    System.out.println("转账失败！余额不足");
                    return false;
                }
            }else{
                System.out.println("转账失败！密码错误");
                return false;
            }
        }
        double getBalance() {
            return balance;
        }

        String getAccountInfo() {
            return "账号是： " + accountNumber + "\n" + "用户是： " + accountHolder + "\n" + "收支结余为： " + balance + "\n" + "您的密码是： " + password;
        }
        public boolean validatePassword(String inputPassword) {
            return true;
        }
        public boolean validateAmount(double amount) {
            return true;
        }
```