

Java06

Task1

为什么 Java 不支持多继承？

我觉得是因为如果子类继承了两个父类，如果两个父类有着**命名相同**的方法，比如一个输出“Hello A”，另一个输出“Hello B”那么子类在调用时程序不知道是到底执行哪个方法，会**报错**。如果想实现多继承这一功能，应该用接口来实现吧，不过也需要在子类重新覆写存在冲突的方法

Task2

Readme:文档里只给了**接口**的代码实现，完整代码包已上传到仓库：)

```
1 package calculus;
2
3 public interface Area { 2 个用法 2 个实现
4     double getS(); 1 个用法 2 个实现
5     double getC(); 1 个用法 2 个实现
6 }
```

Task3

在封装的时候，为了防止类内部的数据被外部**随机访问并修改**，可以用 `private` 关键字保护类内部的属性，保证其不能被外部随意访问，并使用 `setter` 与 `getter` 为外部留出**接口**可以为属性赋值与调用属性的值。`getter` 和 `setter` 在编写时一定要用 `public` 修饰符修饰，以便外部访问。

关于这个名为银行的代码，观察发现他要实现一个银行Atm机的功能系统。

个人信息除了 `balance` (**整错了,想成余额随便看了**) 其他的应该都是隐私不可见

我是先读了各个方法的**返回值类型**来思考这个方法是发挥什么作用的
比如 `deposit` 存款，仅须对该类中的 `balance` 属性操作就行了，不用返回值。
所以代码实现很简单：

```
void deposit(double amount) {  
    this.balance +=amount;  
    System.out.println("存款"+amount+"元! ");  
}
```

浏览中我发现需要实现**判断密码是否相等**，而尾部两个判断取款金额和存款密码的正确性的方法并不完整，便先完成了这两段代码

```
// 只需修改可见性  
public boolean validatePassword(String inputPassword) {  
    if(inputPassword.equals(password))  
    {  
        return true;  
    }  
    return false;  
}  
// 只需修改可见性  
public boolean validateAmount(double amount) {  
    if(amount>this.balance&&amount<0)  
    {  
        return false;  
    }  
    return true;  
}
```

(最简单的是 `getBalance` 方法，只写一行 `return balance` 就行了)

ps: `balance` 也该是 `private`，不然余额随便改也太爽了

我便依次写好了每个方法，并编写了对应的主类验证，并增加了显示信息，增强带入感：)

完整代码包已随此文件一同上传到GitHub仓库里面

下面是主类运行子类可行性运行结果：

"C:\Program Files\Zulu\zulu-24\bin\java.exe" "-;

你好bg欢迎来到微光银行

想干点什么呢？

你好b2g欢迎来到微光银行

想干点什么呢？

查询bg的信息中.....

人类bg持有的卡号36240的银行卡余额还有999999.0元

存款1.0元！

查询bg的信息中.....

人类bg持有的卡号36240的银行卡余额还有1000000.0元

bg正想取钱

密码错了，别想着偷别人的卡就能拿到钱？不可能！

bg正想取钱

bg想取3000000.0元

失败！

有那么多钱吗？

查询b2g的信息中.....

人类b2g持有的卡号37的银行卡余额还有0.0元

bg向卡号37转账100.0元

查询b2g的信息中.....

人类b2g持有的卡号37的银行卡余额还有100.0元