**实 验 报 告 (二)**

|  |  |
| --- | --- |
| **课程名称：** | **Java语言程序设计** |
| **学生姓名：** | **陈立健** |
| **学生学号：** | **201764681171** |
| **学生班级：** | **2017级软件4班** |
| **开课学期：** | **2018-2019学年第一学期** |

**软件学院**

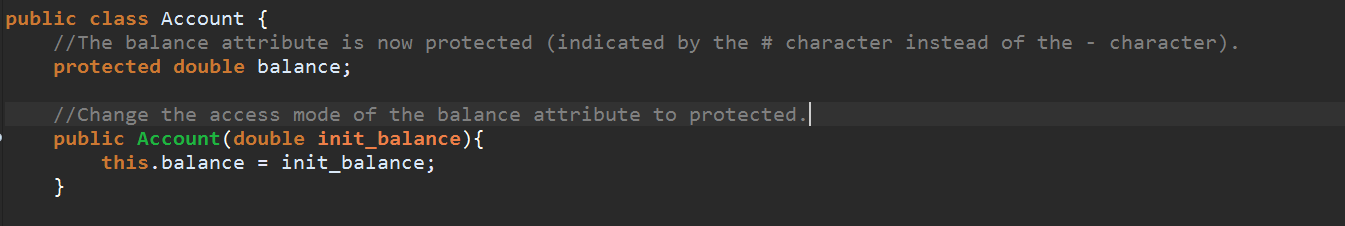
**2018年10月**

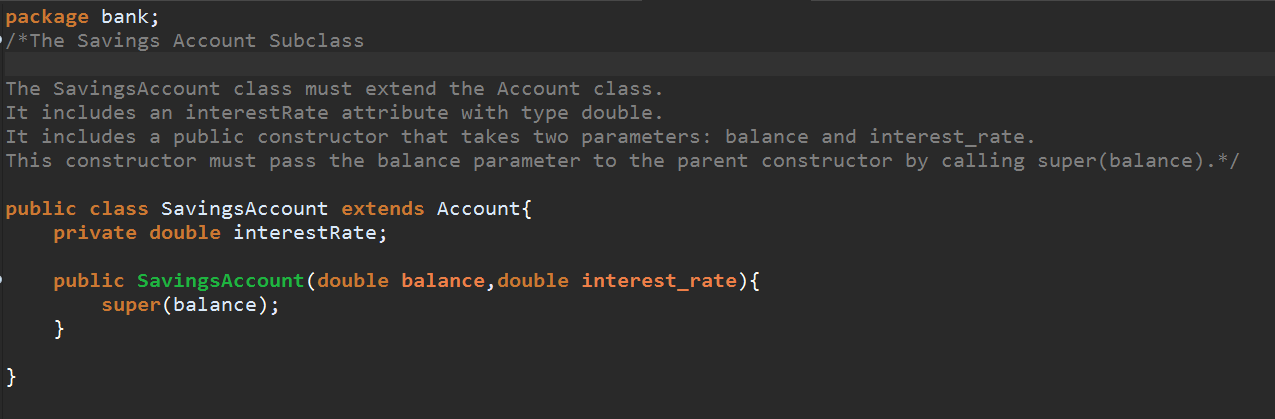
# 实验二 面向对象的编程技术

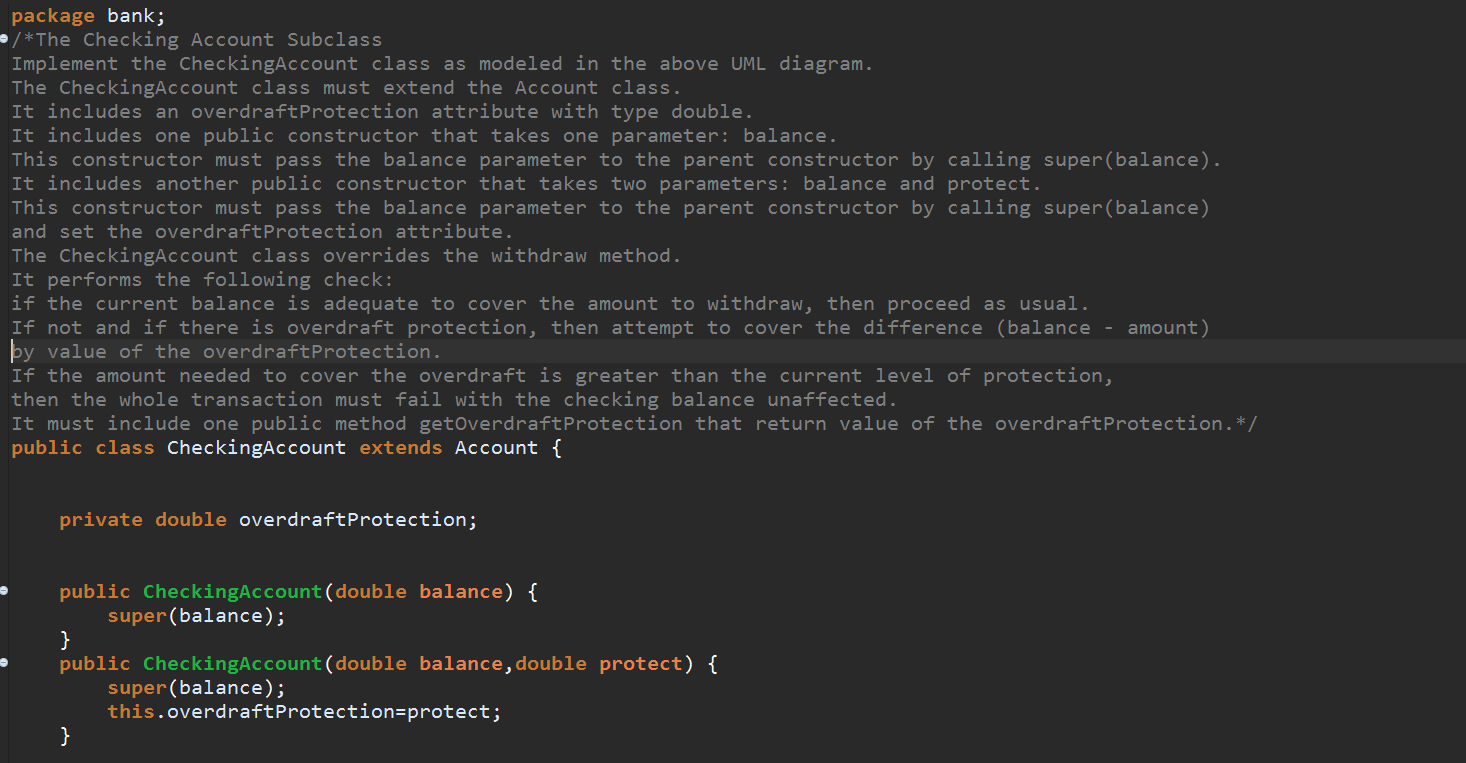
## 代码部分

（注：每个步骤中新建或修改的类的源代码，其中修改的每个部分添加注释，如 /\*\* step 5 ：功能描述 \*\*/）

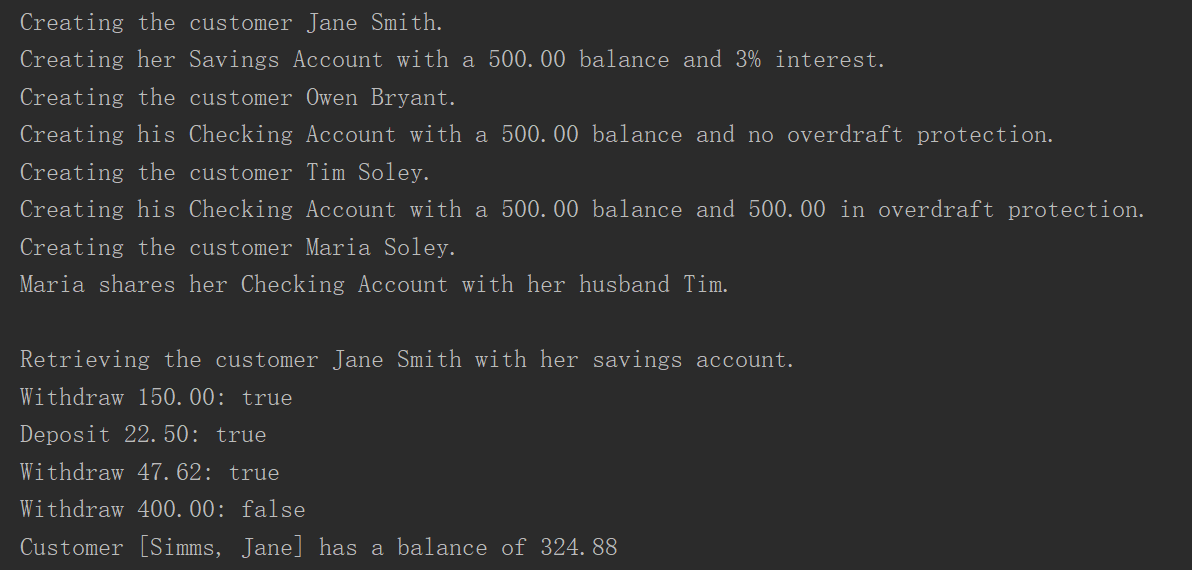
/\*\* step 5 ：功能描述 \*\*/

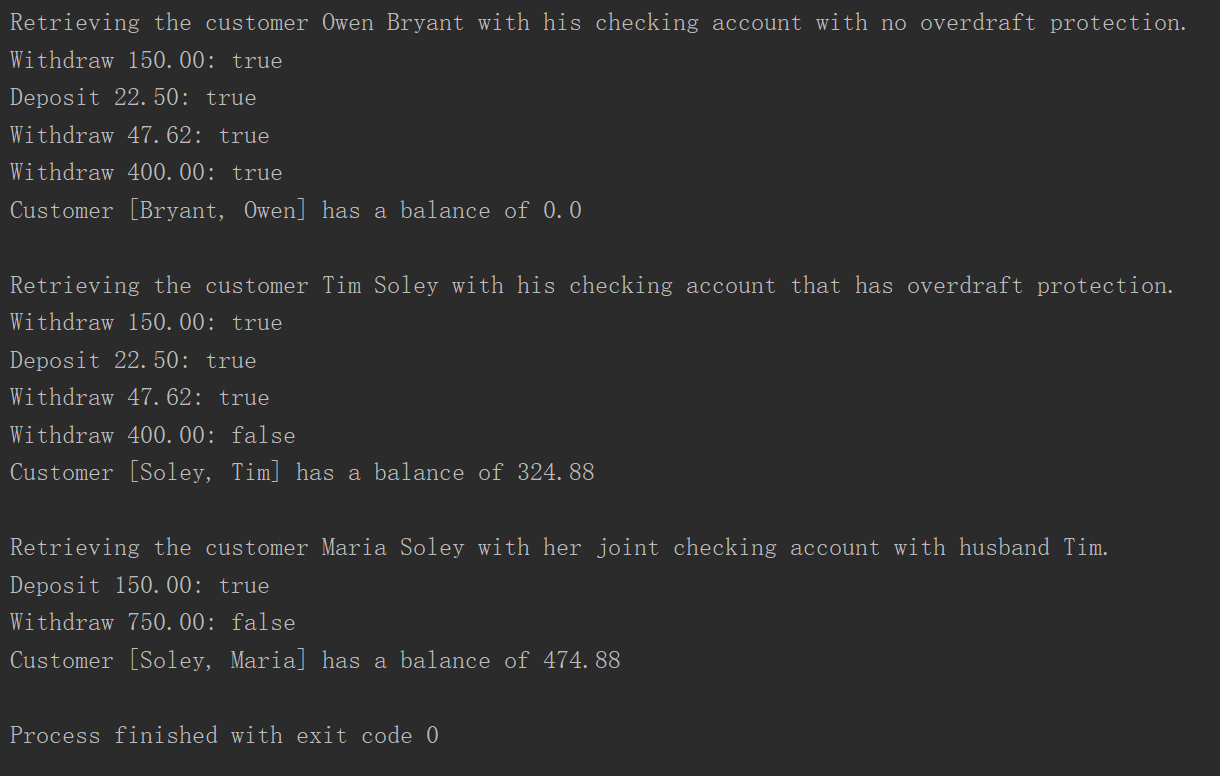




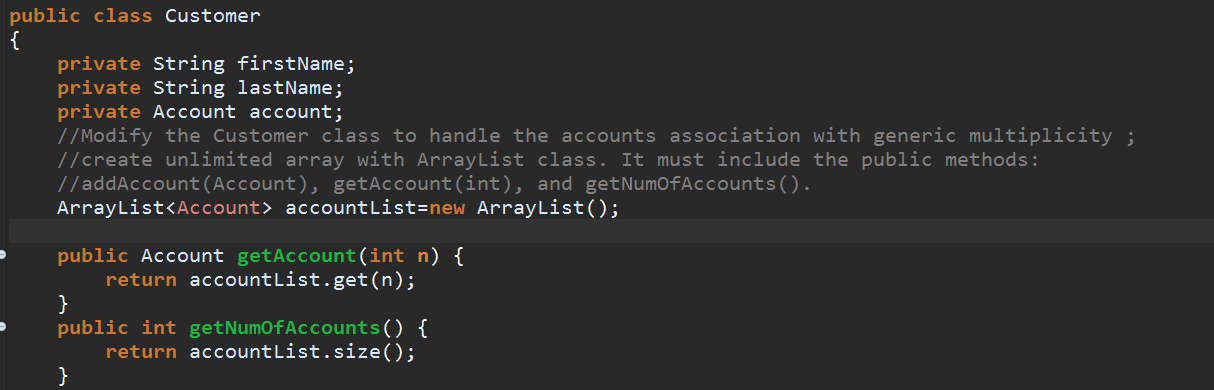


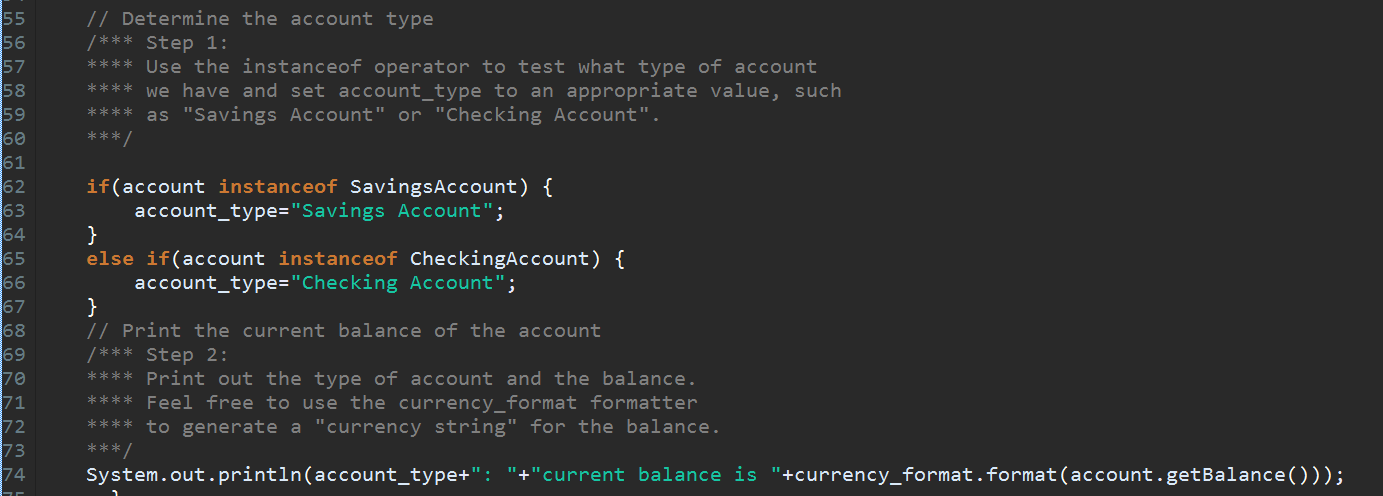






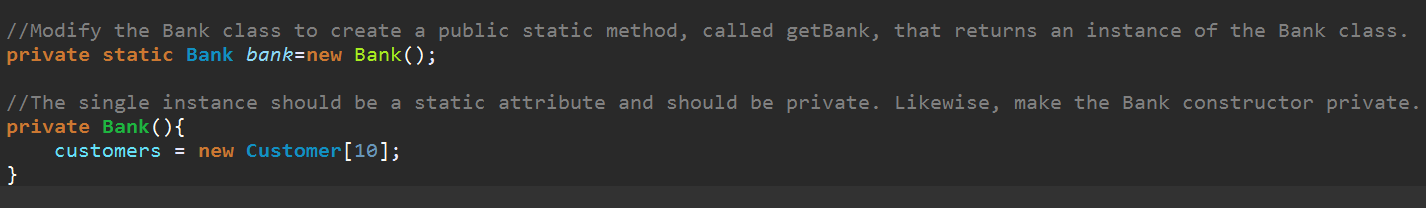
/\*\*STEP6-PRE1 ：功能描述 \*\*/

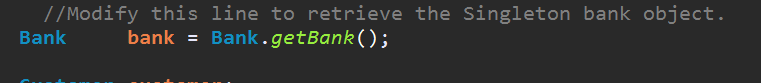


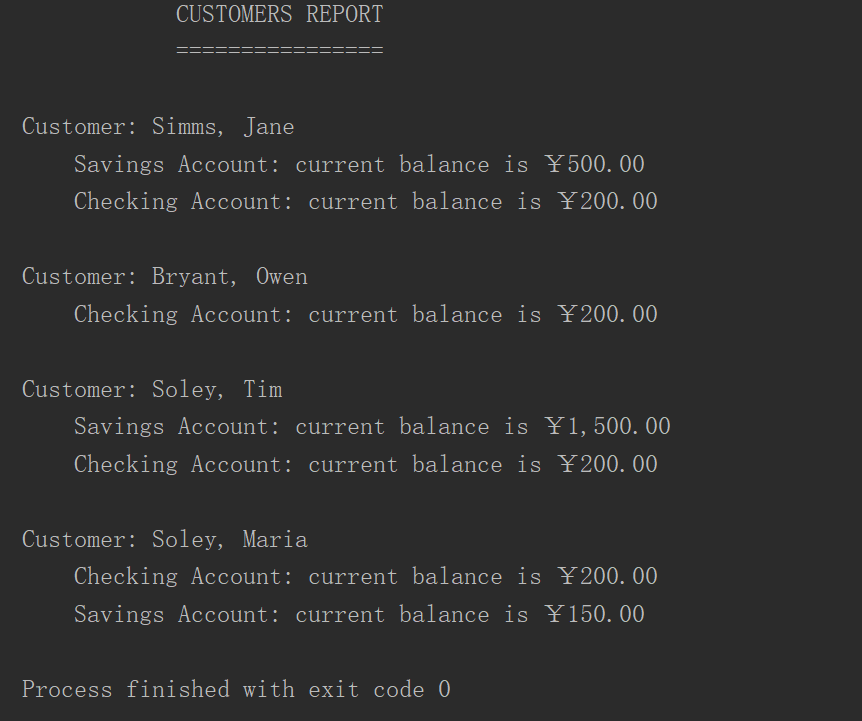




/\*\*STEP6-PRE2 ：功能描述 \*\*/







## 思考问题陈述

1. 任何类只要实现了Comparable接口就可以调用java提供的Array.sort()函数或Collections.sort()函数对一个对象列表排序，在重写Comparable中的函数是可以自定义对列表的排序方式。

如：

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

public class T {

public static void main(String[] args) {

ArrayList list = new ArrayList();

list.add("92.8");

list.add("68.9");

list.add("168.61");

list.add("242");

list.add("317");

list.add("105");

// 字符串排序

Collections.sort(list);

System.out.println(list.toString()); // [105, 168.61, 242, 317, 68.9, 92.8]

Collections.sort(list, new Comparator() {

@Override

public int compare(Object o1, Object o2) {

return new Double((String) o1).compareTo(new Double((String) o2));

}

});

System.out.println(list.toString()); // [68.9, 92.8, 105, 168.61, 242, 317]

}

}

1. Singleton模式主要作用是保证在Java应用程序中，一个类Class只有一个实例存在。

Singleton类的默认构造器被设为私有，这样做可防止其它类使用new关键字直接将对象实例化。对返回Singleton对象的实例方法应用一个静态修饰符，这使得它成为一个类级方法，不创建对象即可进行访问。

当你只需要一个类实例时，Singleton才真正有用；如果类拥有几个实例，使用Singleton就不再适用。

所以，将所有的构造函数都设为private ，而且必须显示的指定构造函数（不能设置为默认的，因为默认构造函数是package访问权限）。