作业 03：设计一个简易银行系统

1. 作业目标

模拟一个简单的用于银行账户管理的系统，实现几个简单业务功能：开户、注销帐号、存款、取款。达到以下目标。

- ✧ 设计简单的类结构；
- ✧ 设计较规范的 Java 程序。
- ✧ 熟练掌握读取键盘或文件输入的基本方法。
- ✧ 重点学习类的职责分工，多个类的合作。

说明：参考下面给出的程序结构来完成，可以扩充和完善。

2. 设计思路

参考实验 3 中的产品管理程序。程序可以初步设计为以下部分：

1) **实体类 AccountInfo**：账户信息封装类，具有属性：accountId（帐号）；password（密码）；balance（余额）等；
2) **业务功能类 BankTransaction**：用于银行业务处理，包括开户、注销帐号、存款、取款等功能函数；同时还应该设计一个属性，用于存放银行账户的数据。
3) **客户端类 Main**：是一个 Java Application 程序，用于和用户交互的客户端类，例如读取用户的输入，以及根据用户的输入来输出相应的信息，即实现账户管理。

说明：这是一个最简化的程序结构，包含了三个部分。随着功能的增加和完善，可以对这三个部分进一步扩充和改进。

3. **Step1**：设计实体类 AccountInfo

将帐户信息封装起来，用一个类 AccountInfo 表示，至少包含以下属性：

```
/** 帐号 */
private String accountId;
/** 密码 */
private String password;
/** 余额 */
private double balance;
```

为了实现对这些私有数据的访问，可以添加相应的 Getter/Setter 方法。

还可以覆盖 toString()方法，根据需要添加一些其他成员。

4. **Step2**：设计银行业务处理类 BankTransaction。

① 新建类 BankTransaction，用于处理银行业务。

- 为存储所管理的账户，需添加一个集合类型数据成员，属性名为 accounts。
- 因为关于账户的信息已经封装为一个类，所以该集合类型的元素类型为 AccoutInfo。

② 设计构造方法，用于创建对象时对数据成员初始化

- 你也可以设计多个构造方法（重载）。

③ 为类 BankTransaction 设计必要的业务操作方法

- 开户帐号：createAccount(……) － 设定新的帐号、密码，余额自然为空啦
- 注销帐号：deleteAccount() － 删除一个账户信息
- 存款：deposit(double money) － 增加余额的值
- 取款：withdraw(double money) －减少余额的值
- 你还可以根据需要，增加其他的方法。

④ 注意：可以合理修改上述步骤、内容或功能，但确保基本功能得以实现。

5. Step3：设计一个客户端程序 BankClient

首先，建立类 BankClient 作为主类，用于处理用户的请求，并执行相应操作。

然后，添加构成这个 Java Application 程序的 main 方法。main 方法的内容，当然是进行银行系统的四种操作（至少），测试是否达到预期目的。由于需要和用户进行交互，根据用户要求进行响应，所以可能包含大量的用户交互操作。

在 Step2 中，有的同学倾向于直接在类 BankTransaction 中添加 main 方法。当功能简单时是可以的。但当系统稍稍复杂或功能增加时，本来是用于银行业务处理的 BankTransaction 类，同时也包含了 main 方法用于一些用户交互操作，如输出菜单、根据用户选择进行处理等。实际上，不应该让一个类承担太多的职责，所以，更好的设计是，把这部分用于用户交互的功能剥离出来，放到一个新的类 BankClient 中。这样，主类（含 main 方法）与业务处理类（含银行操作方法）分离，各自的职责更为独立和明确。

在 main 方法中，具体执行业务操作时，是使用 BankTransaction 来完成的。（创建 BankTransaction 对象，调用其方法，并输出各种结果）

注意：本次作业的程序放在包 *assignment.ass03* 中，可以再设子包。