【案例3-2】银行存取款程序设计

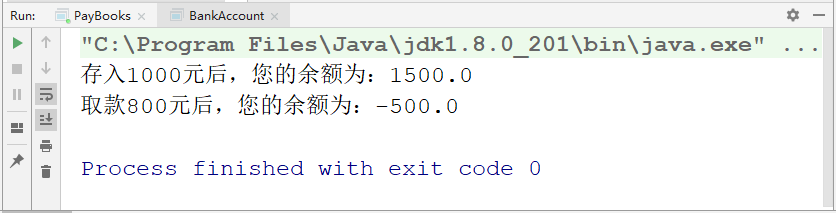
**【案例介绍】**

1. **案例描述**

银行存取款的流程是人们非常熟悉的事情，用户可在银行对自己的资金账户进行存款、取款、查询余额等操作，极大的便利了人民群众对资金的管理。

本任务要求，使用所学知识编写一个银行存取款程序，实现存取款功能。编写一个帐户类实现银行帐户的概念，创建帐户类的对象ba，假设ba的账号为:123456，初始的存款余额为500元。首先向该账户存入1000元，再取出800元。

1. **运行结果**



**【案例目标】**

* 学会分析“银行存取款”程序任务实现的逻辑思路。
* 能够独立完成“银行存取款”程序的源代码编写、运行及编译。
* 理解Java语言是如何体现面向对象编程基本思想的。
* 掌握类的声明以及对象的创建。
* 了解类的成员变量和成员方法的特性以及类的构造方法的使用。

**【案例分析】**

（1）通过任务描述可知，需要定义一个银行帐户类BankAccount实现银行帐户的概念。

（2）账户的属性包括账号和存款余额，所以还需要在BankAccount类中定义两个变量：“帐号” (account\_number) 和“存款余额”(leftmoney)。

（3）对账户的操作包括存款、取款、查询余额，所以再定义四个方法：“存款”(savemoney)、“取款”(getmoney) 、“查询余额”(getleftmoney)以及构造方法(BankAccount)。

（4）最后，编写测试类，在main()方法中创建一个BankAccount类的对象ba，假设ba的账号为:123456，初始的存款余额为500元。首先向该账户存入1000元，再取出2000元。

**【案例实现】**

BankAccount.Java

1. public class BankAccount {
2. int account\_number;//账号
3. double leftmoney;//存款余额
4. public double getleftmoney () {     //查询余额
5. return leftmoney;
6. }
7. public void savemoney(double money) {   //存款
8. leftmoney+=money;
9. }
10. public void getmoney (double money){  //取款
11. leftmoney-=money;
12. }
13. public BankAccount (int number, double money){ //构造方法，用来初始化变量
14. account\_number=number;
15. leftmoney=money;
16. }
17. public static void main(String[] args) {
18. BankAccount ba=new BankAccount(123456,500);
19. ba.savemoney(1000);
20. System.out.println("存入1000元后，您的余额为："+ba.getleftmoney());
21. ba.getmoney(2000);
22. System.out.println("取款800元后，您的余额为："+ba.getleftmoney());
23. }
24. }

上述代码中，声明了一个int类型的account\_number属性设一个double类型的leftmoney属性，并编写了其存款方法savemoney（）、取款方法getmoney（）、查询余额方法getleftmoney（）以及用来初始化变量的构造方法BankAccount（）。最后，在main()方法中进行测试：创建了一个ba对象，账号为123456，初始余额为200元，首先向该账户存入了1000元，随后又取出800元。