《面向对象程序设计与实践》实验报告

实验名称： 《面向对象程序设计与实践》实验十一

学 期： 2022春季学期 任课教师： 陆歌皓 学院 软件学院

专业： 计算机 学号： 20211120138 姓名： 薛凡豪 成绩：

# 一、实验源代码及运行结果：

## 1抽取样本

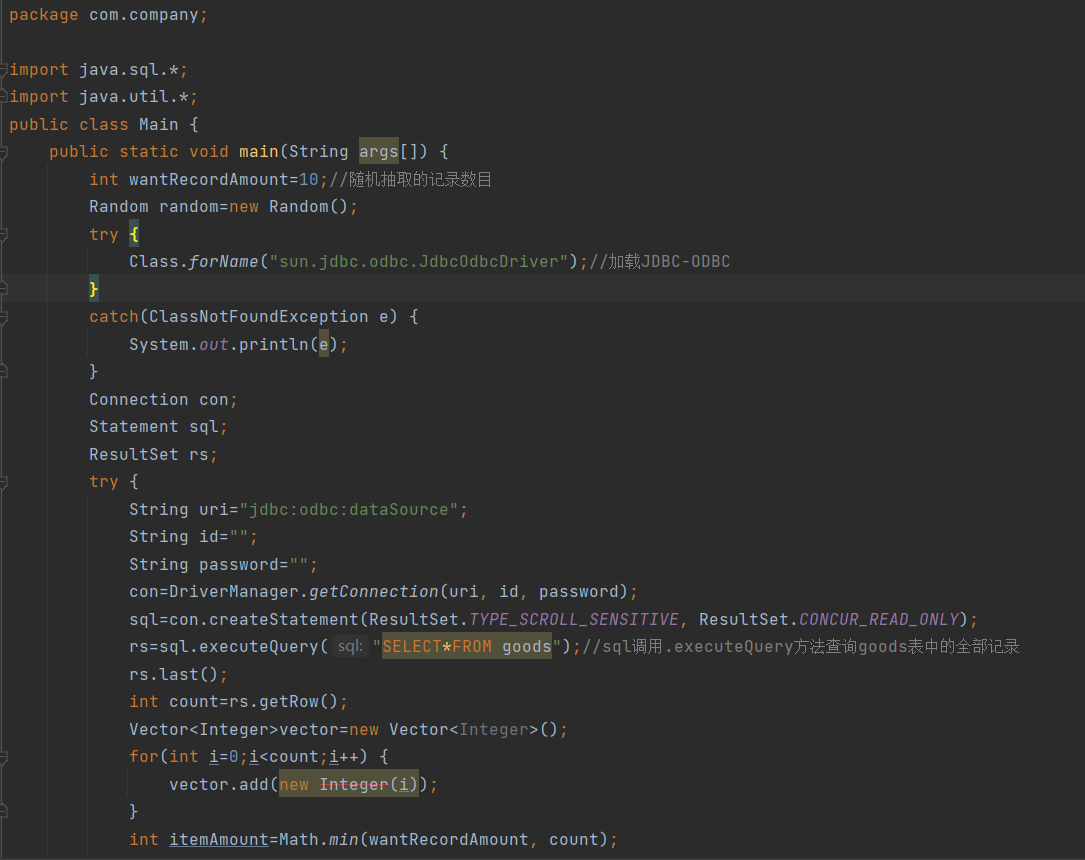
## （1）问题分析：

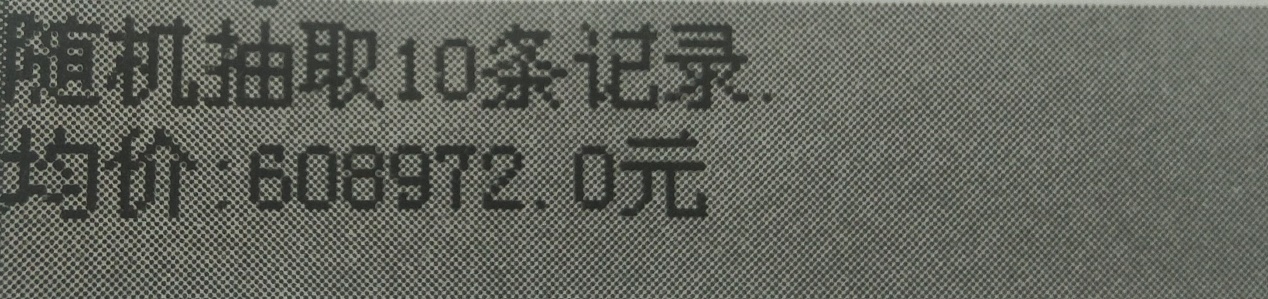
1.本实验主要需要运用的是数据库的链接和使用，即基本步骤。

2.重点是掌握JDBC的使用步骤：与一个数据库链接，向数据库发送SQL语句，处理结果。

3.采用JDBC-ODBC方式访问数据库需要实现设置ODBC数据源。

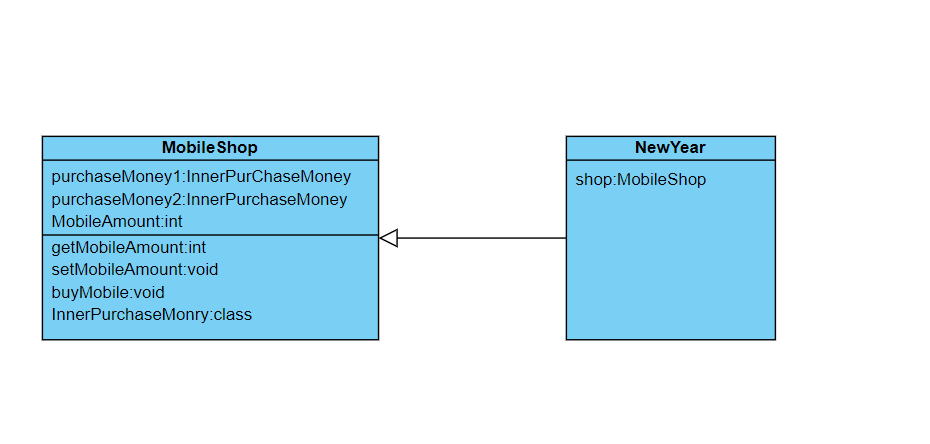
### （2）程序的运行和结果



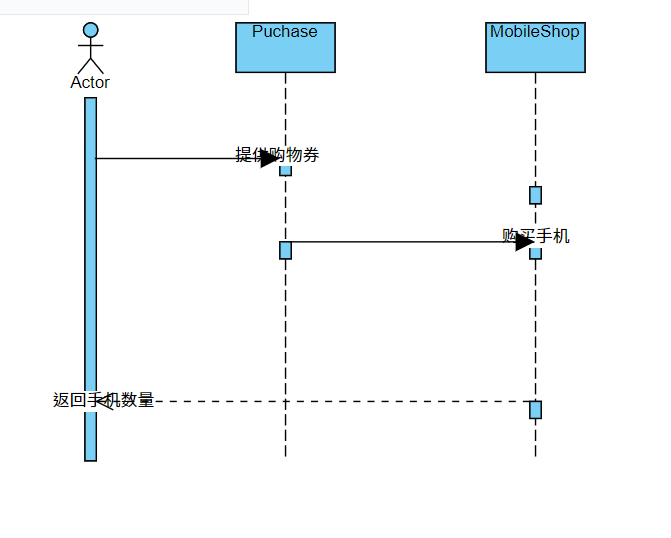


### （3）UML图

类图：



时序图：



### （4）源代码：

11.1  
import java.sql.\*;  
import java.util.\*;  
public class RandomGetRecord {  
 public static void main(String args[]) {  
  int wantRecordAmount=10;//随机抽取的记录数目  
  Random random=new Random();  
  try {  
   Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");//加载JDBC-ODBC  
  }  
  catch(ClassNotFoundException e) {  
   System.out.println(e);  
  }  
  Connection con;  
  Statement sql;  
  ResultSet rs;  
  try {  
   String uri="jdbc:odbc:dataSource";  
   String id="";  
   String password="";  
   con=DriverManager.getConnection(uri, id, password);  
   sql=con.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE, ResultSet.CONCUR\_READ\_ONLY);  
   rs=sql.executeQuery("SELECT\*FROM goods");//sql调用.executeQuery方法查询goods表中的全部记录  
   rs.last();  
   int count=rs.getRow();  
   Vector<Integer>vector=new Vector<Integer>();  
   for(int i=0;i<count;i++) {  
    vector.add(new Integer(i));  
   }  
   int itemAmount=Math.min(wantRecordAmount, count);  
   System.out.println("随机抽取"+itemAmount+"条记录");  
   double sum=0,n=itemAmount;  
   while(itemAmount>0) {  
    int randomIndex=random.nextInt(vector.size());  
    int index=(vector.elementAt(randomIndex)).intValue();  
    //将rs的游标移到index  
    String number=rs.getString(1);  
    String name=rs.getString(2);  
    java.util.Date data=rs.getDate(3);  
    double price=rs.getDouble(4);  
    sum=sum+price;  
    itemAmount--;  
    vector.removeElementAt(randomIndex);  
   }  
   con.close();  
   double aver=sum/n;  
   System.out.println("均价"+aver+"元");  
  }  
  catch(SQLException e) {  
   System.out.println(""+e);  
  }  
 }  
}  
.

## 2.用户转账

### （1）问题分析：

1.本次实验主要练习事务处理问题

2.事务由一组SQL语句组成，即应用程序保证事务中的SQL语句要么全部执行要么一个都不执行。

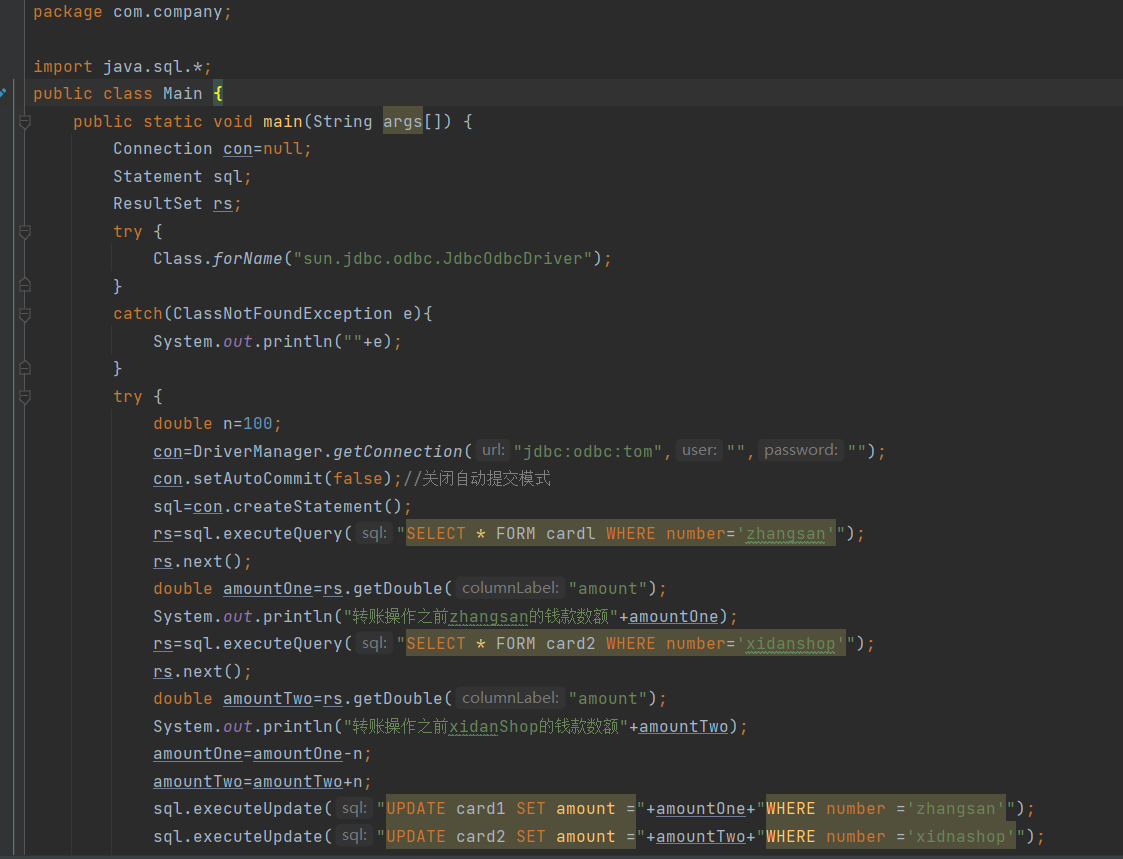
3处理事务的步骤：1：关闭自动提交模式，即关闭SQL语句的即

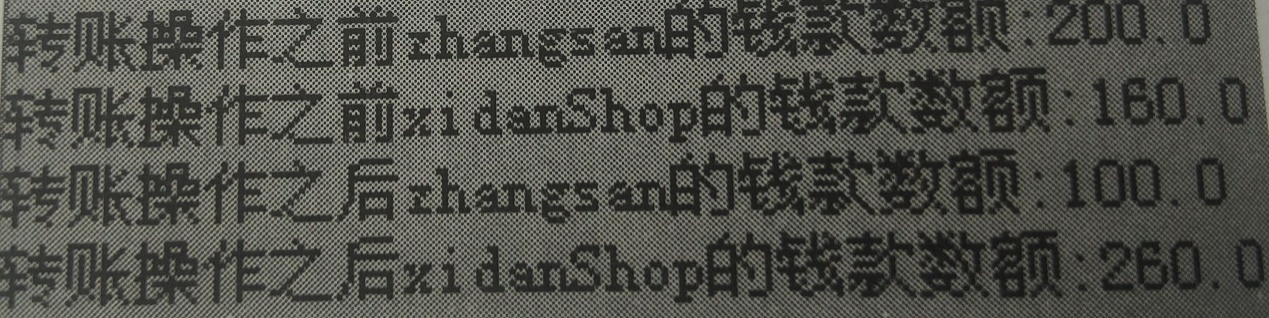
刻生效性。

2：执行事务中的SQL语句，然后执行Conection对象con恢复SQL的有效性。

3：撤销事务所作的操作，即处理事务失败。

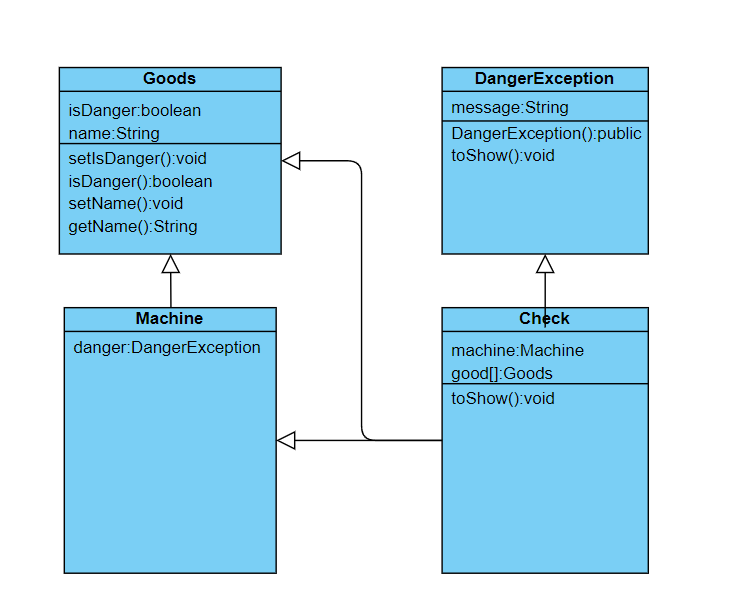
### （2）程序的运行和结果



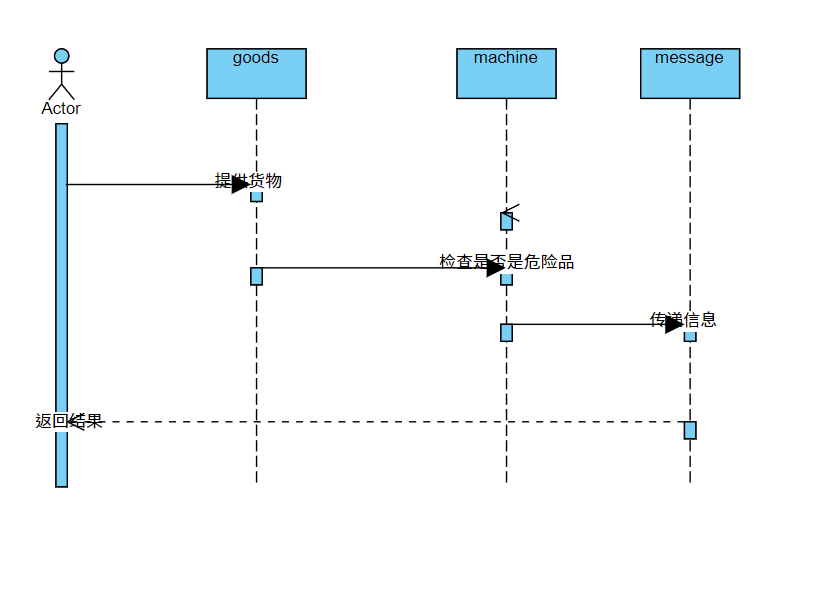


### （3）UML图

类图：



时序图：



### （4）源代码

11.2  
import java.sql.\*;  
public class TurnMoney {  
 public static void main(String args[]) {  
  Connection con=null;  
  Statement sql;  
  ResultSet rs;  
  try {  
   Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  
  }  
  catch(ClassNotFoundException e){  
   System.out.println(""+e);  
  }  
  try {  
   double n=100;  
   con=DriverManager.getConnection("jdbc:odbc:tom","","");  
   con.setAutoCommit(false);//关闭自动提交模式  
   sql=con.createStatement();  
   rs=sql.executeQuery("SELECT \* FORM cardl WHERE number=&apos;zhangsan&apos;");  
   rs.next();  
   double amountOne=rs.getDouble("amount");  
   System.out.println("转账操作之前zhangsan的钱款数额"+amountOne);  
   rs=sql.executeQuery("SELECT \* FORM card2 WHERE number=&apos;xidanshop&apos;");  
   rs.next();  
   double amountTwo=rs.getDouble("amount");  
   System.out.println("转账操作之前xidanShop的钱款数额"+amountTwo);  
   amountOne=amountOne-n;  
   amountTwo=amountTwo+n;  
   sql.executeUpdate("UPDATE card1 SET amount ="+amountOne+"WHERE number =&apos;zhangsan&apos;");  
   sql.executeUpdate("UPDATE card2 SET amount ="+amountTwo+"WHERE number =&apos;xidnashop&apos;");  
   con.commit();  
   con.setAutoCommit(true);//恢复自动提交模式  
   rs=sql.executeQuery("SELECT \* FROM card1 WHERE number =&apos;zhangsan&apos;");  
   rs.next();  
   amountOne=rs.getDouble("amount");  
   System.out.println("转账操作之后zhangsan的钱款数额"+amountOne);  
   rs=sql.executeQuery("SELECT \* FROM card2 WHERE number =&apos;xidanshop&apos;");  
   rs.next();  
   amountTwo=rs.getDouble("amount");  
   System.out.println("转账操作之后zhangsan的钱款数额"+amountTwo);  
   con.close();  
  }  
  catch(SQLException e){  
   try {  
    con.rollback();//撤销事务所做的操作  
   }  
   catch(SQLException exp) {}  
   System.out.println(e.toString());  
  }  
 }  
}}  
}3.查询Excel电子表格

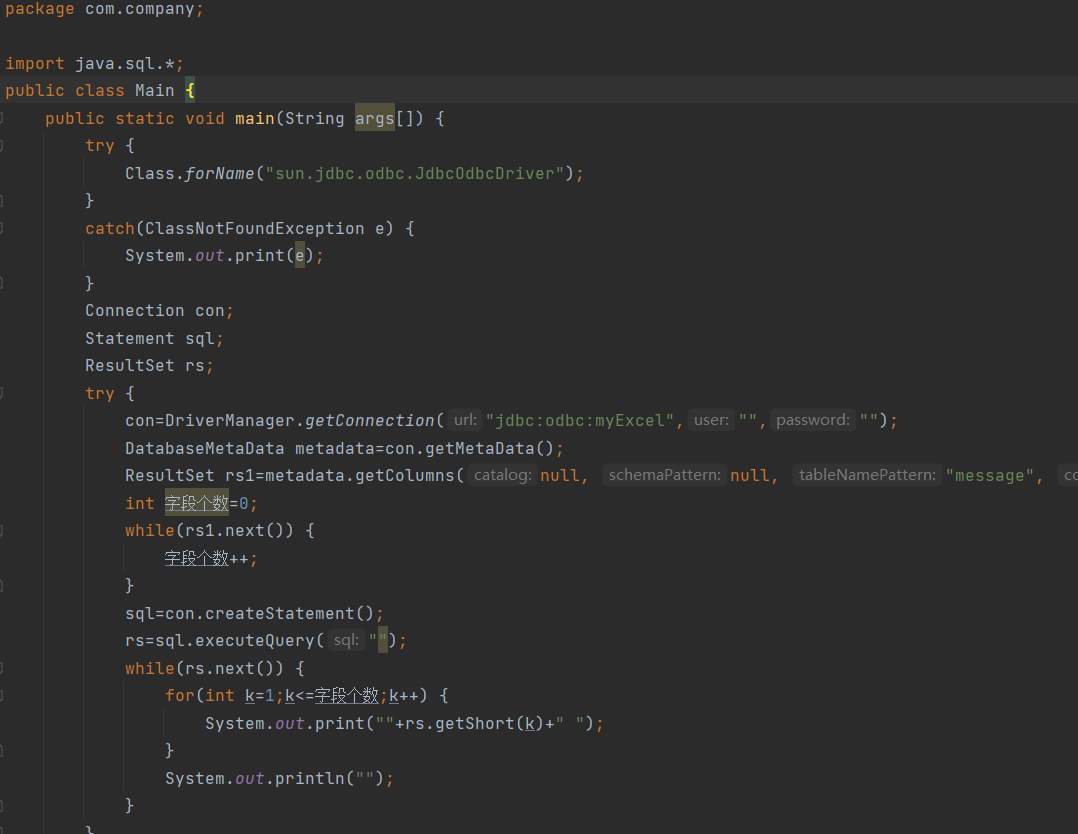
### (1)问题分析：

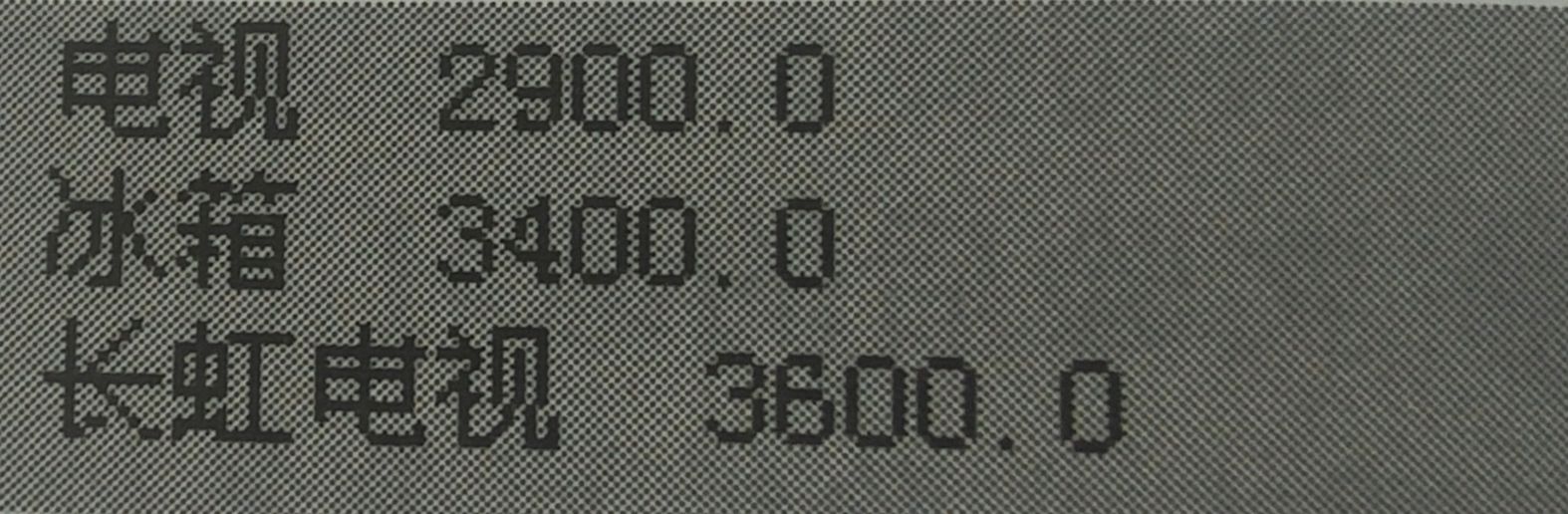
1.主要练习使用掌握操作excel表格的方法。

2.问题：编写一个程序，可以通过JDBC-ODBC桥接器访问excel电子表格，进行操作。

3.本次实验操作excel表格，需要先选定一个工作区作为连接使用的表格。

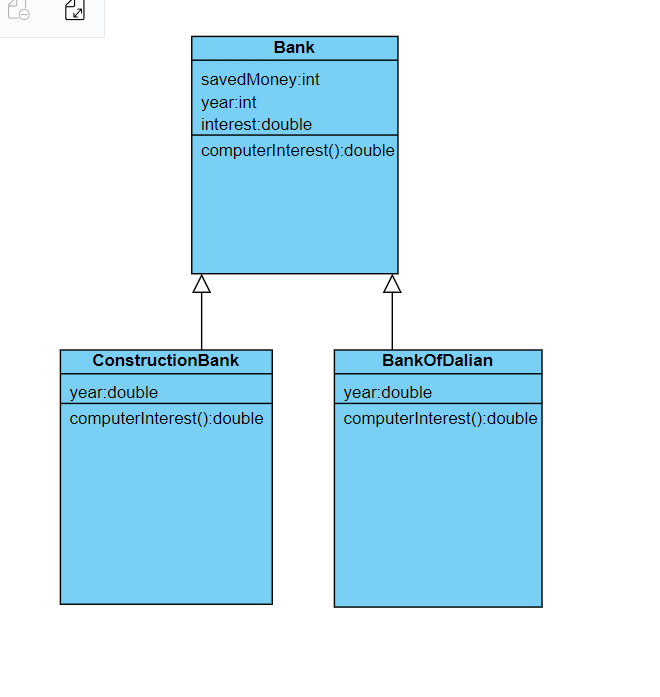
### (2)程序的运行和结果



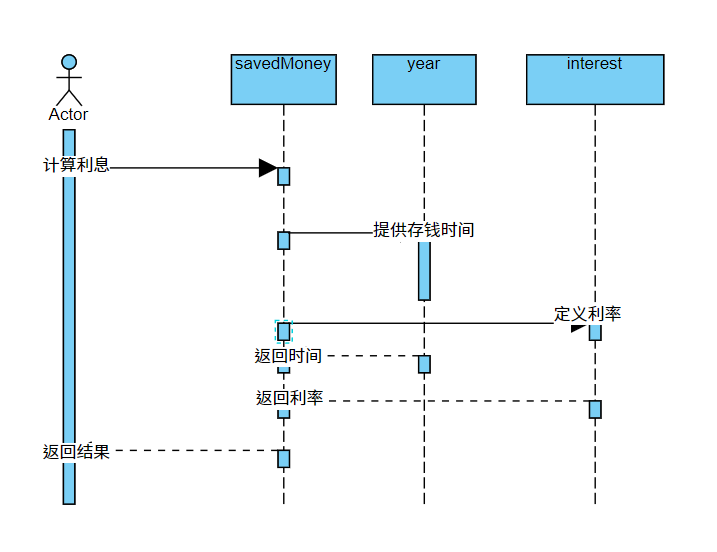


### （3）UML图

类图：



时序图：



### （4）源代码

11.3  
import java.sql.\*;  
public class QueryExcel {  
 public static void main(String args[]) {  
  try {  
   Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  
  }  
  catch(ClassNotFoundException e) {  
   System.out.print(e);  
  }  
  Connection con;  
  Statement sql;  
  ResultSet rs;  
  try {  
   con=DriverManager.getConnection("jdbc:odbc:myExcel","","");  
   DatabaseMetaData metadata=con.getMetaData();  
   ResultSet rs1=metadata.getColumns(null, null, "message", null);  
   int 字段个数=0;  
   while(rs1.next()) {  
    字段个数++;  
   }  
   sql=con.createStatement();  
   rs=sql.executeQuery("");  
   while(rs.next()) {  
    for(int k=1;k<=字段个数;k++) {  
     System.out.print(""+rs.getShort(k)+" ");  
    }  
    System.out.println("");  
   }  
  }  
  catch(SQLException e) {  
   System.out.println(""+e);  
  }  
 }  
}

二、总结

1. 本次实验主要运用了数据库来解决问题。

2.首先使用了JDBC来实现具体的问题。

3. 第二个使用SQL语句进行事务处理，学习使用SQL语句进行事务处理用法来进行数据分析。

4.第三个是JDBC来处理电子表格。

5.本次实验主要运用java的数据库用法，包括JDBC链接数据库，SQL语句进行事务处理，还有电子表格的处理问题。