```
// joi/1/bank/Bank.java
//
//
// Copyright 2003 Bill Campbell and Ethan Bolker

/**
 * A Bank object simulates the behavior of a simple bank/ATM.
 * It contains a Terminal object and two BankAccount objects.
 *
 * Its single public method is open, which opens this Bank
 * for business, prompting the customer for input.
 *
 * To create a Bank and open it for business issue the command
 * <code>java Bank</code>.
 *
 * @see BankAccount
 * @version 1
 */

public class Bank
{
    private String bankName;      // the name of this Bank

    private Terminal atm;         // for talking with the customer

    private BankAccount account1; // two accounts to play with
    private BankAccount account2;

    private static final int INITIAL_BALANCE = 200;
    private static final String HELPSTRING =
        "Transactions: exit, help, deposit, withdraw, balance";

    /**
     * Construct a Bank with the given name.
     * Create two new BankAccounts, each with a starting balance
     * of initialBalance.
     *
     * @param name the name of the Bank.
     */

    public Bank( String name )
    {
        bankName = name;
        atm      = new Terminal();
        account1 = new BankAccount( INITIAL_BALANCE );
        account2 = new BankAccount( INITIAL_BALANCE );
    }

    /**
     * Open the Bank for business.
     *
     * Send a whichAccount message prompting for a BankAccount
     * number, then send a processTransactionsForAccount
     * message to do the work.
     */

```



```
    public void open()
    {
        atm.println( "Welcome to " + bankName );
        boolean bankIsOpen = true;
        while ( bankIsOpen ) {
            BankAccount account = this.whichAccount();
            if ( account == null ) {
                bankIsOpen = false;
            }
            else {
                this.processTransactionsForAccount(account);
            }
        }
        atm.println( "Goodbye from " + bankName );
    }

    // Prompt the user for an account number and return the
    // corresponding BankAccount object. Return null when
    // the Bank is about to close.

    private BankAccount whichAccount()
    {
        int  accountNumber =
            atm.readInt("Account number (1 or 2), 0 to shut down: ");

        if ( accountNumber == 1 ) {
            return account1;
        }
        else if ( accountNumber == 2 ) {
            return account2;
        }
        else if ( accountNumber == 0 ) {
            return null;
        }
        else {
            atm.println( "No account numbered " +
                                accountNumber + "; try again" );
            return this.whichAccount();
        }
    }

    // Prompt the user for transaction to process.
    // Then send an appropriate message to account.

    private void processTransactionsForAccount( BankAccount account)
    {
        atm.println( HELPSTRING );

        boolean moreTransactions = true;
        while ( moreTransactions ) {
            String command = atm.readWord( "transaction: " );
            if ( command.equals( "exit" ) ) {
                moreTransactions = false;
            }
            else if ( command.equals( "help" ) ) {
                atm.println( HELPSTRING );
```

```
113             }
114             else if ( command.equals( "deposit" ) ) {
115                 int amount = atm.readInt( "amount: " );
116                 account.deposit( amount );
117             }
118             else if ( command.equals( "withdraw" ) ) {
119                 int amount = atm.readInt( "amount: " );
120                 account.withdraw( amount );
121             }
122             else if ( command.equals( "balance" ) ) {
123                 atm.println( account.getBalance() );
124             }
125             else{
126                 atm.println("sorry, unknown transaction" );
127             }
128         }
129     }
130
131     /**
132      * The Bank simulation program begins here when the user
133      * issues the command <code>java Bank</code>.
134      *
135      * @param args the command line arguments (ignored).
136      */
137
138     public static void main( String[] args )
139     {
140         Bank javaBank = new Bank( "Engulf and Devour" );
141         javaBank.open();
142     }
143 }
```