/** A BankAccount object has private fields to keep track of its current balance, the number of transactions performed and the Bank in which it is an account, and public methods to access those fields appropriately. 

* @see Bank
* @version 4 */

public class BankAccount
{
private int balance = 0;  // Account balance (whole dollars)
private int transactionCount = 0;  // Number of transactions performed.
private Bank issuingBank;  // Bank issuing this account

/** Construct a BankAccount with the given initial balance and issuing Bank. Construction counts as this BankAccount's first transaction.

* @param initialBalance the opening balance.
* @param issuingBank the bank that issued this account.
*/

public BankAccount( int initialBalance, Bank issuingBank )
{
this.issuingBank = issuingBank;
deposit( initialBalance );
}

/** Withdraw the given amount, decreasing this BankAccount's balance and the issuing Bank's balance. Counts as a transaction.

* @param amount the amount to be withdrawn
* @return amount withdrawn
*/

public int withdraw( int amount )
{
incrementBalance( -amount );
countTransaction();
return amount ;
}

/** Deposit the given amount, increasing this BankAccount's balance and the issuing Bank's balance. Counts as a transaction.

* @param amount the amount to be deposited
* @return amount deposited
*/

public int deposit(int amount)
{
incrementBalance( amount);
countTransaction();
return amount ;
}

/** Request for balance. Counts as a transaction.

* @return current account balance
*/

public int requestBalance()
{
countTransaction();
return getBalance() ;
}

/** Get the current balance.
* Does NOT count as a transaction.

* @return current account balance
*/

public int getBalance()
{
return balance;
}

/** Increment account balance by given amount. Also increment issuing Bank's balance. Does NOT count as a transaction.

* @param amount the amount increment.
*/

public void incrementBalance( int amount )
{
balance += amount;
this.getIssuingBank().incrementBalance( amount );
}

/** Get the number of transactions performed by this account. Does NOT count as a transaction.

* @return number of transactions performed.
*/

public int countTransactions()
{
return transactionCount;
}

private void countTransaction() { transactionCount++ ;
}

private Bank getIssuingBank() { return issuingBank ;
}

private void deposit( int amount ) { balance += amount ;
}

private void incrementBalance( int amount ) { balance += amount ;
getIssuingBank().incrementBalance( amount ) ;
}

private int getBalance() { return balance ;
}

private int getTransactionCount() { return transactionCount ;
}

private void setTransactionCount( int count ) { transactionCount = count ;
}}
public int getTransactionCount() {
    return transactionCount;
}

/**
 * Increment by 1 the count of transactions, for this account
 * and for the issuing Bank.
 * Does NOT count as a transaction.
 * Get the bank that issued this account.
 */

public void countTransaction() {
    transactionCount++;
    this.getIssuingBank().countTransaction();
}

/**
 * Get the bank that issued this account.
 * Does NOT count as a transaction.
 *
 * @return issuing bank.
 */

public Bank getIssuingBank() {
    return issuingBank;
}