/**
 * A SavingsAccount is a BankAccount that bears interest.
 * A fee is charged for too many transactions in a month.
 *
 * @see BankAccount
 *
 * @version 7
 */

public class SavingsAccount extends BankAccount {

private int transactionsThisMonth;

/**
 * Override getTransactionFee() to return a non-zero fee
 * after the appropriate number of free monthly transactions.
 *
 * @return the fee for current transaction.
 */

protected int getTransactionFee() {
    if (transactionsThisMonth >
        getIssuingBank().getMaxFreeTransactions()) {
        return getIssuingBank().getTransactionFee();
    } else {
        return 0;
    }
}

/**
 * Increment count of transactions, for this account for
 * this Month and in total  and for the issuing Bank, By one.
 *
 * @exception InsufficientFundsException when appropriate.
 */

public void countTransaction()
    throws InsufficientFundsException {
    transactionsThisMonth++;
    super.countTransaction();
}

/**
 * Constructor, accepting an initial balance.
 * @param initialBalance the opening balance.
 *
 * @param issuingBank the bank that issued this account.
 *
 * @exception InsufficientFundsException when appropriate.
 */

public SavingsAccount( int initialBalance, Bank issuingBank) {
    super( initialBalance, issuingBank);
    transactionsThisMonth = 1;
}

/**
 * A SavingsAccount earns interest each month.
 *
 * @exception InsufficientFundsException when appropriate.
 */

public void newMonth()
    throws InsufficientFundsException {
    double monthlyRate = getIssuingBank().getInterestRate()/12;
    incrementBalance( (int)(monthlyRate * getBalance()));
    transactionsThisMonth = 0;
}
}