```
 1  // joi/7/bank/SavingsAccount.java
 2  //
 3  //
 4  // Copyright 2003 Bill Campbell and Ethan Bolker
 5
 6  /**
 7   * A SavingsAccount is a BankAccount that bears interest.
 8   * A fee is charged for too many transactions in a month.
 9   *
10   * @see BankAccount
11   *
12   * @version 7
13   */
14
15  public class SavingsAccount extends BankAccount
16  {
17      private int transactionsThisMonth;
18
19      /**
20       * Override getTransactionFee() to return a non-zero fee
21       * after the appropriate number of free monthly transactions.
22       *
23       * @return the fee for current transaction.
24       */
25
26      protected int getTransactionFee()
27      {
28          if (transactionsThisMonth >
29              getIssuingBank().getMaxFreeTransactions()) {
30              return getIssuingBank().getTransactionFee();
31          }
32          else {
33              return 0;
34          }
35      }
36
37      /**
38       * Increment count of transactions, for this account for
39       * this Month and in total  and for the issuing Bank, by one.
40       *
41       * @exception InsufficientFundsException when appropriate.
42       */
43
44      public void countTransaction()
45          throws InsufficientFundsException
46      {
47          transactionsThisMonth++;
48          super.countTransaction();
49      }
50
51      /**
52       * Constructor, accepting an initial balance.
53       * @param initialBalance the opening balance.
54       *
55       * @param issuingBank the bank that issued this account.
56       *
```



```
57       * @exception InsufficientFundsException when appropriate.
58       */
59
60      public SavingsAccount( int initialBalance, Bank issuingBank )
61           throws InsufficientFundsException
62      {
63          super( initialBalance, issuingBank);
64          transactionsThisMonth = 1;
65      }
66
67      /**
68       * A SavingsAccount earns interest each month.
69       *
70       * @exception InsufficientFundsException when appropriate.
71       */
72
73      public void newMonth()
74           throws InsufficientFundsException
75      {
76          double monthlyRate = getIssuingBank().getInterestRate()/12;
77          incrementBalance( (int)(monthlyRate * getBalance()));
78          transactionsThisMonth = 0;
79      }
80  }
```