```
 1  // joi/7/bank/CheckingAccount.java
 2  //
 3  //
 4  // Copyright 2003 Bill Campbell and Ethan Bolker
 5
 6  /**
 7   * A CheckingAccount is a BankAccount with one new feature:
 8   * the ability to cash a check by calling the honorCheck method.
 9   * Each honored check costs the customer a checkFee.
10   *
11   * @see BankAccount
12   *
13   * @version 7
14   */
15
16  public class CheckingAccount extends BankAccount
17  {
18      /**
19       * Constructs a CheckingAccount with the given
20       * initial balance and issuing Bank.
21       * Counts as this account's first transaction.
22       *
23       * @param initialBalance the opening balance for this account.
24       * @param issuingBank the bank that issued this account.
25       *
26       * @exception InsufficientFundsException when appropriate.
27       */
28
29      public CheckingAccount( int initialBalance, Bank issuingBank )
30          throws InsufficientFundsException
31      {
32          super( initialBalance, issuingBank );
33      }
34
35      /**
36       * Honor a check:
37       * Charge the account the appropriate fee
38       * and withdraw the amount.
39       *
40       * @param amount amount (in whole dollars) to be withdrawn.
41       * @return the amount withdrawn.
42       *
43       * @exception InsufficientFundsException when appropriate.
44       */
45
46      public int honorCheck( int amount )
47          throws InsufficientFundsException
48      {
49          // careful error checking logic:
50          // first try to deduct the check fee
51          // if you succeed, try to honor check
52          // if that fails, remember to add back the check fee!
53
54          try {
55              incrementBalance( - getIssuingBank().getCheckFee() );
56          }
```



```
57          catch (InsufficientFundsException e) {
58              throw new InsufficientFundsException(
59                          "to cover check fee" );
60          }
61          try {
62              withdraw( amount );
63          }
64          catch (InsufficientFundsException e) {
65              incrementBalance( getIssuingBank().getCheckFee() );
66              throw new InsufficientFundsException(
67                          "to cover check + check fee" );
68          }
69          return amount;
70      }
71
72      /**
73       *  Nothing special happens to a CheckingAccount on the
74       *  first day of the month.
75       */
76
77      public void newMonth()
78      {
79          return;
80      }
81  }
```