```
 1  // joi/4/estore/EStore.java
 2  //
 3  //
 4  // Copyright 2003 Bill Campbell and Ethan Bolker
 5
 6  /**
 7   * An EStore object simulates the behavior of a simple on line
 8   * shopping web site.
 9   *
10   * It contains a Terminal object to model the customer's browser
11   * and a Catalog of Items that may be purchased and
12   * then added to the customer's shoppingCart.
13   *
14   * @version 4
15   */
16
17  public class EStore
18  {
19      private String   storeName;
20      private Terminal browser;
21      private Catalog  catalog;
22
23      /**
24       *  Construct a new EStore.
25       *
26       *  @param storeName the name of the EStore
27       *  @param browser the visitor's Terminal.
28       */
29
30      public EStore( String storeName, Terminal browser )
31      {
32          this.browser   = browser;
33          this.storeName = storeName;
34          this.catalog   = new Catalog();
35          catalog.addItem( new Item("quaffle", 55) );
36          catalog.addItem( new Item("bludger", 15) );
37          catalog.addItem( new Item("snitch",  1000) );
38      }
39
40      /**
41       *  Visit this EStore.
42       *
43       *  Execution starts here when the store opens for
44       *  business. User can visit as a customer, act as
45       *  the manager, or exit.
46       */
47
48      public void visit()
49      {
50          // Print a friendly welcome message.
51          browser.println( "Welcome to " + storeName );
52          while (true) { // an infinite loop ...
53              browser.println();
54              String whoAreYou = browser.readWord(
55                 storeName + " (manager, visit, exit): ");
56              if (whoAreYou.equals("exit")) {
```



```
57                  break; // leave the while loop
58              }
59              if (whoAreYou.equals("manager")) {
60                  managerVisit();
61              }
62              if (whoAreYou.equals("visit")) {
63                  customerVisit();
64              }
65          }
66      }
67
68      /**
69       *  Manager options:
70       *
71       *  examine the catalog
72       *  add an Item to the catalog
73       *  quit
74       */
75      private void managerVisit( )
76      {
77          while (true) {
78              String cmd =
79                browser.readWord("manager command (show, new, quit):");
80              if (cmd.equals("quit")) {
81                  break;  // leave manager command while loop
82              }
83              else if (cmd.equals("show")) {
84                  catalog.show(browser);
85              }
86              else if (cmd.equals("new")) {
87                  String itemName = browser.readWord("  item name:     ");
88                  int cost        = browser.readInt( "  cost:         ");
89                  catalog.addItem( new Item(itemName, cost) );
90              }
91              else {
92                  browser.println("unknown manager command: " + cmd);
93              }
94          }
95      }
96
97      /**
98       * Customer visits this EStore.
99       *
100      * Loop allowing customer to select items to add to her
101      * shoppingCart.
102      */
103
104     private void customerVisit( )
105     {
106         // Create a new, empty ShoppingCart.
107         ShoppingCart basket = new ShoppingCart();
108
109         browser.println( "Currently available:");
110         catalog.show(browser);
111         while ( true ) { // loop forever ...
112             String nextPurchase = browser.readWord(
```

```
                    "select your purchase, checkout, help: ");

            if ( nextPurchase.equals("checkout" )) break; // leave loop!

            if ( nextPurchase.equals("help" )) {
                catalog.show(browser);
                continue; // go back to top of while loop
            }
            // customer has entered the name of an Item
            basket.addItem( catalog.getItem(nextPurchase) );
        }

        int numberPurchased = basket.getCount();
        browser.println("We are shipping these " +
                        basket.getCount() + " Items:");
        basket.showContents(browser);
        browser.println("and charging your account $" + basket.getCost())
        browser.println("Thank you for shopping at " + storeName);
    }

    /**
     *  The EStore simulation program begins here when the user
     *  issues the command <code>java EStore</code>
     *
     *  If first command line argument is "-e" instantiate a
     *  Terminal that echoes its input.
     *
     *  The next command line argument (if there is one)
     *  is the name of the EStore.
     *
     *  @param args <-e> <storeName>
     */
    public static void main( String[] args )
    {

        String storeName = "Virtual Minimal Minimall"; //default

        // check to see if first argument is "-e"
        boolean echo = ( (args.length > 0) && (args[0].equals("-e")) );

        // if first argument was "-e" then look at second for store name
        int nextArg = (echo ? 1 : 0 );

        if (args.length > nextArg) {
            storeName = args[nextArg];
        }

        // Print this to simulate internet search.
        System.out.println("connecting ...");

        // Create an EStore object and visit it
        (new EStore(storeName, new Terminal(echo))).visit();
    }
}
```