

```

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

```

```
113         "select your purchase, checkout, help: ");
114
115         if ( nextPurchase.equals("checkout" )) break; // leave loop!
116
117         if ( nextPurchase.equals("help" )) {
118             catalog.show(browser);
119             continue; // go back to top of while loop
120         }
121         // customer has entered the name of an Item
122         basket.addItem( catalog.getItem(nextPurchase) );
123     }
124
125     int numberPurchased = basket.getCount();
126     browser.println("We are shipping these " +
127         basket.getCount() + " Items:");
128     basket.showContents(browser);
129     browser.println("and charging your account $" + basket.getCost())
130     browser.println("Thank you for shopping at " + storeName);
131 }
132
133 /**
134  * The EStore simulation program begins here when the user
135  * issues the command <code>java EStore</code>
136  *
137  * If first command line argument is "-e" instantiate a
138  * Terminal that echoes its input.
139  *
140  * The next command line argument (if there is one)
141  * is the name of the EStore.
142  *
143  * @param args <-e> <storeName>
144  */
145     public static void main( String[] args )
146     {
147
148         String storeName = "Virtual Minimal Minimal"; //default
149
150         // check to see if first argument is "-e"
151         boolean echo = ( (args.length > 0) && (args[0].equals("-e")) );
152
153         // if first argument was "-e" then look at second for store name
154         int nextArg = (echo ? 1 : 0 );
155
156         if (args.length > nextArg) {
157             storeName = args[nextArg];
158         }
159
160         // Print this to simulate internet search.
161         System.out.println("connecting ...");
162
163         // Create an EStore object and visit it
164         (new EStore(storeName, new Terminal(echo))).visit();
165     }
166 }
```