```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using SwinAdventure;
7  using System.Xml.Linq;
8  using System.ComponentModel;
9  using System.Formats.Tar;
10
11 namespace SwinAdventure
12 {
13     public class LookCommand : Command  //Change from internal to public
14     {
15         //Step 1 of the LookCommand.cs in the UML design
16         public LookCommand() : base(new string[] { "look" })
17         {
18
19         }
20         //Step 2 of the LookCommand.cs in the UML design
21         public override string Execute(Player p, string[] text)
22         {
23             IHaveInventory _container = null;
24             string _itemid;
25             string _containerid;
26             //Check the array text for the length
27             if (text.Length != 3 && text.Length != 5)
28             {
29                 return "I don't know how to look like that";
30             }
31             //If the first word must be “look”, return “Error in look
                 input”
32             if (text[0].ToLower() != "look")
33             {
34                 return "Error in look input";
35             }
36             //The second word must be “at”, otherwise return “What do you
                 want to look at?”
37             if (text[1].ToLower() != "at")
38             {
39                 return "What do you want to look at?";
40             }
41             //If there are 5 elements, then the 4th word must be “in”,
                 otherwise return “What do you want to look in?”
42             if (text.Length == 5)
43             {
44                 if (text[3].ToLower() != "in")
45                 {
46                     return "What do you want to look in?";
```

```
47                  }
48                  _containerid = text[4]; // "backpack"
49                  _container = FetchContainer(p, _containerid); // This gets
                      the Bag object
50              }
51              //If there are 3 elements, the container is the player
52              if (text.Length == 3)
53              {
54                  _container = p;
55              }
56
57              _itemid = text[2]; // "map"
58              return LookAtIn(_itemid, _container);
59          }
60          //Step 3 of the LookCommand.cs in the UML design
61          private IHaveInventory FetchContainer(Player p, string containerId)
62          {
63              if (p.AreYou(containerId)) // "me" or "inventory" should
                  identify the player
64              {
65                  return p;
66              }
67              // Otherwise, try to locate it in the player's inventory (for
                  bags, etc.)
68              return p.Locate(containerId) as IHaveInventory;
69          }
70
71          ////Step 4 of the LookCommand.cs in the UML design
72          private string LookAtIn(string thingId, IHaveInventory container)
73          {
74              if (container == null)
75              {
76                  return "I cannot find the " + thingId;
77              }
78              GameObject item = container.Locate(thingId); // This calls
                  Bag.Locate("map")
79
80              if (item == null)
81              {
82                  return "I cannot find the " + thingId + " in the " +
                      container.Name;
83              }
84              return item.FullDescription;
85          }
86      }
87  }
```