```csharp
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Iteration5
8  {
9      public class LookCommand : Command
10     {
11         public LookCommand() : base(new string[] { "look" })
12         {
13         }
14
15         public override string Execute(Player p, string[] text)
16         {
17             if ((text.Length != 3) && (text.Length != 5))
18             {
19                 return "I don't know how to look like that";
20             }
21             else if (text[0] != "look")
22             {
23                 return "Error in look input";
24             }
25             else if (text[1] != "at")
26             {
27                 return "What do you want to look at?";
28             }
29
30             if ((text.Length == 5) && (text[3] != "in"))
31             {
32                 return "What do you want to look in?";
33             }
34
35             String itemId = text[2];
36             IhaveInventory container = p;
37
38             if (text.Length == 5)
39             {
40                 container = FetchContainer(p, text[4]);
41                 if (container == null)
42                 {
43                     return $"I cannot find the {text[4]}";
44                 }
45             }
46
47             return LookAtIn(itemId, container);
48         }
49
```

```csharp
50          private IhaveInventory FetchContainer(Player p, string containerId)
51          {
52              return p.Locate(containerId) as IhaveInventory;
53          }
54
55          private string LookAtIn(string thingId, IhaveInventory container)
56          {
57              var item = container.Locate(thingId);
58              if (item != null)
59              {
60                  return item.FullDescription;
61              }
62              else
63              {
64                  return $"I can't find the {thingId}";
65              }
66          }
67      }
68  }
69
```