

```
1 using swinadventure;
2
3 public class Bag : Item, IHaveInventory
4 {
5     private Inventory _inventory;
6
7     public Bag(string[] ids, string name, string desc)
8         : base(ids, name, desc)
9     {
10         _inventory = new Inventory();
11     }
12
13     public override string FullDescription
14     {
15         get
16         {
17             return $"In the {Name} you see:\n{_inventory.ItemList}";
18         }
19     }
20
21     public Inventory Inventory
22     {
23         get { return _inventory; }
24     }
25
26     public GameObject Locate(string id)
27     {
28         if (AreYou(id))
29         {
30             return this;
31         }
32
33         return _inventory.Fetch(id);
34     }
35 }
36
```

```
1 using System;
2
3 public abstract class Command : IdentifiableObject
4 {
5     public Command(string[] ids) : base(ids) { }
6
7     public abstract string Execute(Player p, string[] text);
8 }
9
```

```
1 public interface IHaveInventory
2 {
3     GameObject Locate(string id);
4     string Name { get; }
5 }
6
```

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

```
50
51
52     private string LookAtIn(string thingId, GameObject container)
53     {
54         IHaveInventory containerWithInventory = container as IHaveInventory;
55
56         if (containerWithInventory == null)
57         {
58             return $"{container.Name} is not a container";
59         }
60
61         GameObject item = containerWithInventory.Locate(thingId);
62
63         if (item != null)
64         {
65             return item.FullDescription;
66         }
67
68         if (container == containerWithInventory)
69         {
70             return $"I cannot find the {thingId}";
71         }
72
73         return $"I cannot find the {thingId} in the {container.Name}";
74     }
75 }
76
77
78
```

```
1 using NUnit.Framework;
2
3 [TestFixture]
4 public class LookCommandTests
5 {
6     private Player _player;
7     private Item _gem;
8     private Item _sword;
9     private Bag _bag;
10    private LookCommand _lookCommand;
11
12    [SetUp]
13    public void Setup()
14    {
15        _player = new Player("Ved Jay Makhijani", "104762184");
16        _gem = new Item(new string[] { "gem" }, "a gem", "A shiny gem");
17        _sword = new Item(new string[] { "sword" }, "a sword", "A sharp sword");
18        _bag = new Bag(new string[] { "bag" }, "a leather bag", "A large leather bag");
19        _lookCommand = new LookCommand();
20
21        _player.Inventory.Put(_gem);
22        _player.Inventory.Put(_bag);
23        _bag.Inventory.Put(_sword);
24    }
25
26    [Test]
27    public void TestLookAtMe()
28    {
29        string result = _lookCommand.Execute(_player, new string[]
30        { "look", "at", "inventory" });
31        string expected = "You are Ved Jay Makhijani, 104762184\nYou are carrying:\n\t" + "a gem (gem)\n\t" + "a leather bag (bag)";
32        Assert.AreEqual(expected, result);
33    }
34
35    [Test]
36    public void TestLookAtGem()
37    {
38        string result = _lookCommand.Execute(_player, new string[]
39        { "look", "at", "gem" });
40        Assert.AreEqual("A shiny gem", result);
41    }
42
43    [Test]
44    public void TestLookAtUnk()
45    {
46        string result = _lookCommand.Execute(_player, new string[]
```

```
        { "look", "at", "ring" });
45     Assert.AreEqual("I cannot find the ring", result);
46 }
47
48 [Test]
49 public void TestLookAtGemInInventory()
50 {
51     string result = _lookCommand.Execute(_player, new string[]
52         { "look", "at", "gem", "in", "inventory" });
53     Assert.AreEqual("A shiny gem", result);
54 }
55
56 [Test]
57 public void TestLookAtGemInBag()
58 {
59     string result = _lookCommand.Execute(_player, new string[]
60         { "look", "at", "sword", "in", "bag" });
61     Assert.AreEqual("A sharp sword", result);
62 }
63
64 [Test]
65 public void TestLookAtItemInNoBag()
66 {
67     string result = _lookCommand.Execute(_player, new string[]
68         { "look", "at", "sword", "in", "box" });
69     Assert.AreEqual("I cannot find the box", result);
70 }
71
72 [Test]
73 public void TestLookAtNoGemInBag()
74 {
75     string result = _lookCommand.Execute(_player, new string[]
76         { "look", "at", "gem", "in", "bag" });
77     Assert.AreEqual("I cannot find the gem", result);
78 }
79
80 [Test]
81 public void TestInvalidLook()
82 {
83     string result1 = _lookCommand.Execute(_player, new string[]
84         { "look", "around" });
85     Assert.AreEqual("I don't know how to look like that", result1);
86 }
```

```
1 public class Player : GameObject, IHaveInventory
2 {
3     private Inventory _inventory;
4
5     public Player(string name, string desc) : base(new string[] { "me",
6         "inventory" }, name, desc)
7     {
8         _inventory = new Inventory();
9     }
10    public Inventory Inventory
11    {
12        get { return _inventory; }
13    }
14
15    public override string FullDescription
16    {
17        get
18        {
19            return $"You are {Name}, {base.FullDescription}\nYou are
20                carrying:\n{_inventory.ItemList}";
21        }
22    }
23
24    public GameObject Locate(string id)
25    {
26        if (AreYou(id))
27        {
28            return this;
29        }
30        return _inventory.Fetch(id);
31    }
32 }
33
34
```


