

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace IdentifiableObject
8 {
9     public class Bag:Item, IHaveInventory
10    {
11        private Inventory _inventory;
12
13        public Bag(string[] ids, string name, string desc): base(ids, name,
14            desc)
15        {
16            _inventory = new Inventory();
17        }
18
19        public GameObject Locate(string id)
20        {
21            if(AreYou(id)==true)
22            {
23                return this;
24            }
25            else
26            {
27                return _inventory.Fetch(id);
28            }
29            return null;
30        }
31
32        public string FullDescription
33        {
34            get
35            {
36                return $"In the {this.Name} you can see:\n
37                    {_inventory.ItemList}";
38            }
39        }
40
41        public Inventory Inventory
42        {
43            get
44            {
45                return _inventory;
46            }
47        }
48    }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Xml.Linq;
7
8 namespace IdentifiableObject
9 {
10     public class Player:GameObject, IHaveInventory
11     {
12         private Inventory _inventory;
13
14         public Player(string name, string desc) : base(new string[] {"me", ↗
15             "inventory"}, name, desc)
16         {
17             _inventory = new Inventory();
18         }
19
20         public GameObject Locate(string id)
21         {
22             if (AreYou(id) == true)
23             {
24                 return this;
25             }
26             else
27             {
28                 return _inventory.Fetch(id);
29             }
30         }
31
32         public override string FullDescription
33         {
34             get
35             {
36                 return $"You are ({Name}), ({base.FullDescription}). You ↗
37                     are carrying:\n{_inventory.ItemList}";
38             }
39         }
40
41         public Inventory Inventory
42         {
43             get
44             {
45                 return _inventory;
46             }
47         }
48     }
49 }
```

```
48 }  
49
```

```
1 using System;  
2 using System.Collections.Generic;  
3 using System.Linq;  
4 using System.Text;  
5 using System.Threading.Tasks;  
6  
7 namespace IdentifiableObject  
8 {  
9     public interface IHaveInventory  
10     {  
11         public GameObject Locate(string id);  
12  
13         public string Name  
14         {  
15             get;  
16         }  
17     }  
18 }  
19
```

```
1 using System;  
2 using System.Collections.Generic;  
3 using System.Linq;  
4 using System.Text;  
5 using System.Threading.Tasks;  
6  
7 namespace IdentifiableObject  
8 {  
9     public abstract class Command:IdentifiableObject  
10     {  
11         public Command(string[] ids):base(ids)  
12         {  
13  
14         }  
15  
16         public abstract string Execute(Player p, string[] text);  
17     }  
18 }  
19
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace IdentifiableObject
8 {
9     public class LookCommand:Command
10    {
11        public LookCommand():base(new string[] {"look"})
12        {
13
14        }
15
16        public override string Execute(Player p, string[] text)
17        {
18            if(text.Length==3 || text.Length==5)
19            {
20                if (text[0].ToLower() == "look")
21                {
22                    if (text[1].ToLower() == "at")
23                    {
24                        if(text.Length==3)
25                        {
26                            return LookAtIn(text[2], p);
27                        }
28                        else if(text.Length==5)
29                        {
30                            if (text[3].ToLower() == "in")
31                            {
32                                return LookAtIn(text[2], FetchContainer(p,
33                                text[4]));
34                            }
35                            else
36                            {
37                                return "What do you want to look in?";
38                            }
39                        }
40                        else
41                        {
42                            return "I don't know how to look like that";
43                        }
44                    }
45                    else return "What do you want to look at?";
46                }
47            }
48            else
49            {
50                return "Error in look input";
51            }
52        }
53    }
54 }
```

```
49         }
50     }
51     else
52     {
53         return "I don't know how to look like that";
54     }
55 }
56
57 private IHaveInventory? FetchContainer(Player p, string containerId)
58 {
59     return p.Locate(containerId) as IHaveInventory;
60 }
61
62 private string LookAtIn(string thingId, IHaveInventory container)
63 {
64     if(container.Locate(thingId)!=null)
65     {
66         return container.Locate(thingId).FullDescription;
67     }
68     return "I can't find the " + thingId;
69 }
70 }
71 }
72
```

```
1 namespace IdentifiableObject
2 {
3     public class Tests
4     {
5         private Player _player;
6         private Player _player_no_bag;
7         private Item _gem;
8         private Bag _bag;
9         private Command _look;
10
11         [SetUp]
12         public void Setup()
13         {
14             _look = new LookCommand();
15             _player = new Player("Duc Thang", "Student");
16             _player_no_bag = new Player("player", "participant");
17             _gem = new Item(new string[] { "gem" }, "a gem", "This is a gem");
18             _bag = new Bag(new string[] { "bag" }, "Thang's bag", "student");
19             _player.Inventory.Put(_bag);
20         }
21
22         [Test]
23         public void TestLookAtMe()
24         {
25             string look_execution = _look.Execute(_player, new string[]
26                 { "look", "at", "inventory" });
27             string output = _player.FullDescription;
28             Assert.AreEqual(look_execution, output);
29         }
30
31         [Test]
32         public void TestLookAtGem()
33         {
34             _player.Inventory.Put(_gem);
35             string look_execution = _look.Execute(_player, new string[]
36                 { "look", "at", "gem" });
37             string output = _gem.FullDescription;
38             Assert.AreEqual(look_execution, output);
39         }
40
41         [Test]
42         public void TestLookAtUnk()
43         {
44             string look_execution = _look.Execute(_player, new string[]
45                 { "look", "at", "gem" });
46             string output = "I can't find the gem";
47             Assert.AreEqual(look_execution, output);
```

```
45     }
46
47     [Test]
48     public void TestLookAtGemInMe()
49     {
50         _player.Inventory.Put(_gem);
51         string look_execution = _look.Execute(_player, new string[] { "look", "at", "gem", "in", "me" });
52         string output = _gem.FullDescription;
53         Assert.AreEqual(look_execution, output);
54     }
55
56     [Test]
57     public void TestLookAtGemInBag()
58     {
59         _bag.Inventory.Put(_gem);
60         string look_execution = _look.Execute(_player, new string[] { "look", "at", "gem", "in", "bag" });
61         string output = _gem.FullDescription;
62         Assert.AreEqual(look_execution, output);
63     }
64
65     [Test]
66     public void TestLookAtGemInNoBag()
67     {
68         string look_execution = _look.Execute(_player_no_bag, new string[] { "look", "at", "bag" });
69         string output = "I can't find the bag";
70         Assert.AreEqual(look_execution, output);
71     }
72
73     [Test]
74     public void TestLookAtNoGemInBag()
75     {
76         string look_execution = _look.Execute(_player, new string[] { "look", "at", "gem", "in", "bag" });
77         string output = "I can't find the gem";
78         Assert.AreEqual(look_execution, output);
79     }
80
81     [Test]
82     public void TestInvalidLook()
83     {
84         Assert.AreEqual(_look.Execute(_player, new string[] { "look", "around" }), "I don't know how to look like that");
85         Assert.AreEqual(_look.Execute(_player, new string[] { "hello", "104776473" }), "I don't know how to look like that");
86         Assert.AreEqual(_look.Execute(_player, new string[] { "look", "at", "Nguyen Duc Thang" }), "I can't find the Nguyen Duc
```

```
    Thang");  
87     }  
88     }  
89 }
```

Test Explorer

Test run finished: 8 Tests (8 Passed, 0 Failed, 0 Skipped) run in 129 ms

0 Warnings 0 Errors

Test	Duration	Traits
BagUnitTest (6)		
IdentifiableObjectUnitTest (7)		
InventoryUnitTest (5)		
ItemUnitTest (4)		
LookCommandUnitTest (8)	14 ms	
IdentifiableObject (8)	14 ms	
Tests (8)	14 ms	
TestInvalidLook	5 ms	
TestLookAtGem	< 1 ms	
TestLookAtGemInBag	9 ms	
TestLookAtGemInMe	< 1 ms	
TestLookAtGemInNoBag	< 1 ms	
TestLookAtMe	< 1 ms	
TestLookAtNoGemInBag	< 1 ms	
TestLookAtUnk	< 1 ms	
PlayerUnitTest (5)		

Run | Debug

**Group Summary**

LookCommandUnitTest

Tests in group: 8

Total Duration: 14 ms

**Outcomes**

8 Passed