

```
1
2
3
4     public class Location : GameObject, IHaveInventory
5     {
6
7         private Inventory _inventory;
8
9         public Location(string name, string description) : base(new string [] { "room", "here" }, name, description)
10        {
11            _inventory = new Inventory();
12        }
13
14        public GameObject Locate(string id)
15        {
16
17            if (AreYou(id))
18            {
19                return this;
20            }
21            else if (_inventory.HasItem(id))
22            {
23                return _inventory.Fetch(id);
24            }
25
26            else
27            {
28                return null;
29            }
30        }
31        public override string FullDescription
32        {
33            get
34            {
35                return $"You are in {Name}\n{FullDescription}\nIn this room you can see:\n{_inventory.ItemList}";
36            }
37        }
38    }
39    public Inventory Inventory
40    {
41        {
42            get
43            {
44                return _inventory;
45            }
46        }
47    }
```

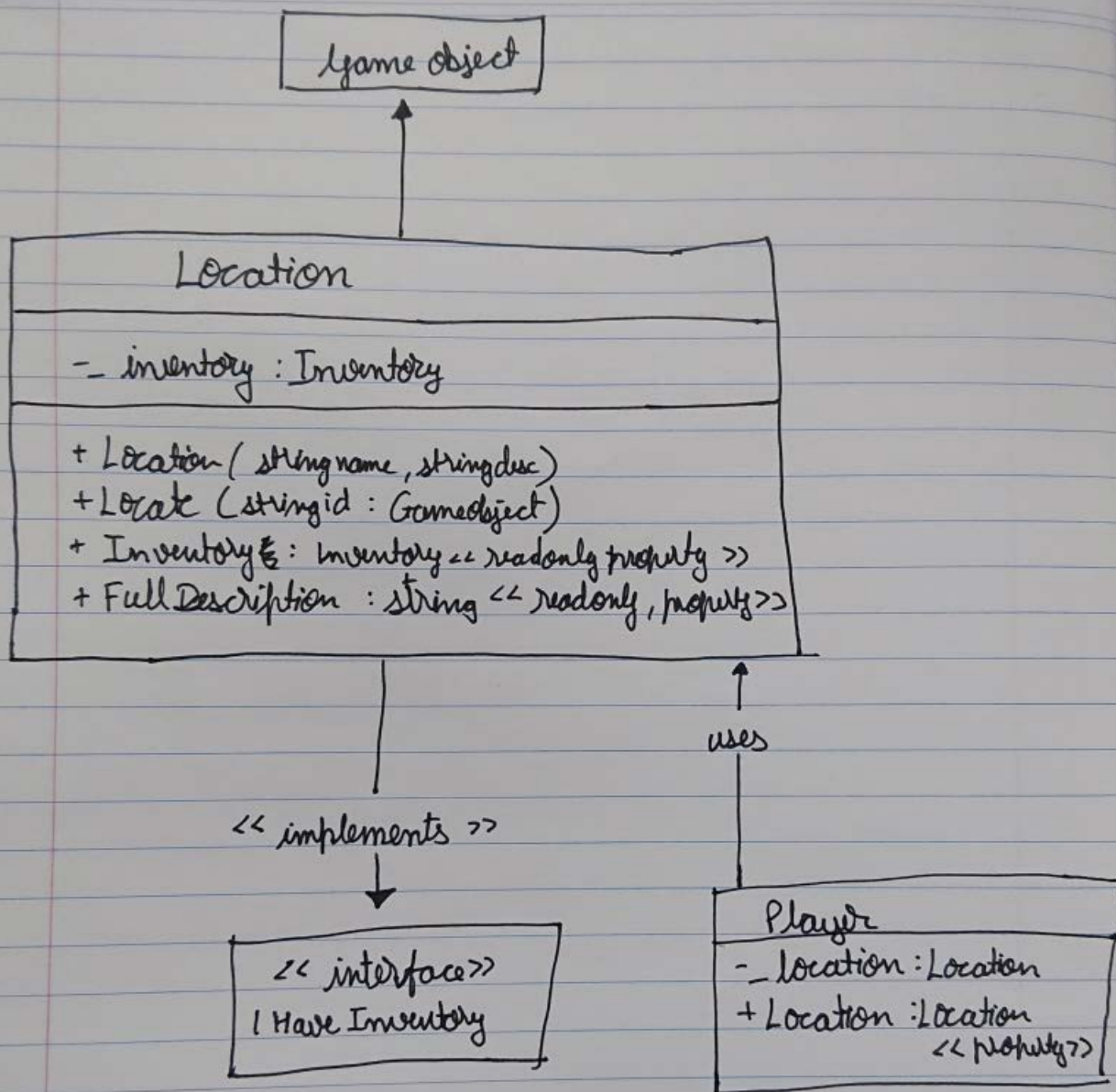
```
48     public Item FetchItem(string id)
49     {
50         return _inventory.Fetch(id);
51     }
52
53     string IHaveInventory.Name
54     {
55         get
56         {
57             return Name;
58         }
59     }
60 }
61
62
```

```
1 using NUnit.Framework;
2
3 [TestFixture]
4 public class TestLocation
5 {
6     private Location location;
7     private Player player;
8     private Item sword;
9
10    [SetUp]
11    public void Setup()
12    {
13
14        location = new Location("a conflict", "In World");
15        player = new Player("Ved", "the student");
16        sword = new Item(new string[] { "sword" }, "a Sword", "a sharp
17        Sword");
18
19        location.Inventory.Put(sword);
20
21
22        player.Location = location;
23    }
24
25    // Test 1: Location can identify itself
26    [Test]
27    public void TestLocationCanIdentifyItself()
28    {
29
30        Assert.That(location.Locate("room"), Is.SameAs(location));
31
32
33        Assert.That(location.Locate("here"), Is.SameAs(location));
34    }
35
36    // Test 2: Location can locate items it contains
37    [Test]
38    public void TestLocationCanLocateItemInInventory()
39    {
40
41        Assert.That(location.Locate("sword"), Is.SameAs(sword));
42    }
43
44    // Test 3: Player can locate items in their location
45
46    [Test]
47    public void TestPlayerCanLocateItemInLocation()
48    {
```

```
49      // Player should be able to locate the sword in the location's inventory
50      Assert.That(player.Locate("sword"), Is.SameAs(sword));
51  }
52
53
54 }
55
```

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

```
50     private GameObject FetchContainer(Player p, string containerId)
51     {
52         return p.Locate(containerId);
53     }
54
55
56     private string LookAtIn(string thingId, GameObject container)
57     {
58         IHaveInventory containerWithInventory = container as IHaveInventory;
59
60         if (containerWithInventory == null)
61         {
62             return $"{container.Name} is not a container";
63         }
64
65         GameObject item = containerWithInventory.Locate(thingId);
66
67         if (item != null)
68         {
69             return item.FullDescription;
70         }
71
72
73         if (container == containerWithInventory)
74         {
75             return $"I cannot find the {thingId}";
76         }
77
78         return $"I cannot find the {thingId} in the {container.Name}";
79     }
80
81 }
82
```



Test Explorer

35

35

0

Search (Ctrl+I)

Test run finished: 3 Tests (3 Passed, 0 Failed, 0 Skipped) run in 250 ms

0 Warnings

0 Errors

Test

Dur

Run

Debug

IdentifiableObjectTests (35)

<Empty Namespace> (35)

BagTests (5)

IdentifiableObjectTests (7)

InventoryTests (4)

ItemTests (3)

LookCommandTests (8)

PlayerTests (5)

TestLocation (3)

TestLocationCanIdentify...

TestLocationCanLocateIt...

TestPlayerCanLocateIt...

Test Detail Summary

TestPlayerCanLocateItemInLocation

Source: [locationstest.cs](#) line 47

Duration: < 1 ms



