```
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 _6._1P
9  {
10     public class Bag : Item, IHaveInventory
11     {
12         private Inventory _inventory;
13
14         public Bag(string[] ids, string name, string description) : base
             (ids, name, description)
15         {
16             _inventory = new Inventory();
17         }
18
19         public GameObject Locate(string id)
20         {
21             if (AreYou(id))
22             {
23                 return this;
24             }
25             return _inventory.Fetch(id);
26         }
27
28         public override string FullDescription
29         {
30             get
31             {
32                 return $"In the {Name} you can see: {string.Join(", ",
                     _inventory.ItemList)}"; //add "," between every
                     elements
33             }
34         }
35
36         public Inventory Inventory
37         { get { return _inventory; } }
38
39
40
41     }
42 }
43
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace _6._1P
8  {
9      public abstract class Command : IdentifiableObject //base class for
         other classes, cannot create an object
10     {
11         private string[] _ids;
12         public Command(string[] ids) : base(ids)
13         {
14             _ids = ids;
15         }
16
17         public abstract string Execute(Player p, string[] text); //
             define without implementation
18
19
20
21     }
22 }
23
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace _6._1P
8  {
9      public interface IHaveInventory
10     {
11         GameObject Locate(string id); //locate item
12         string Name { get; } //a name property
13     }
14 }
15
```

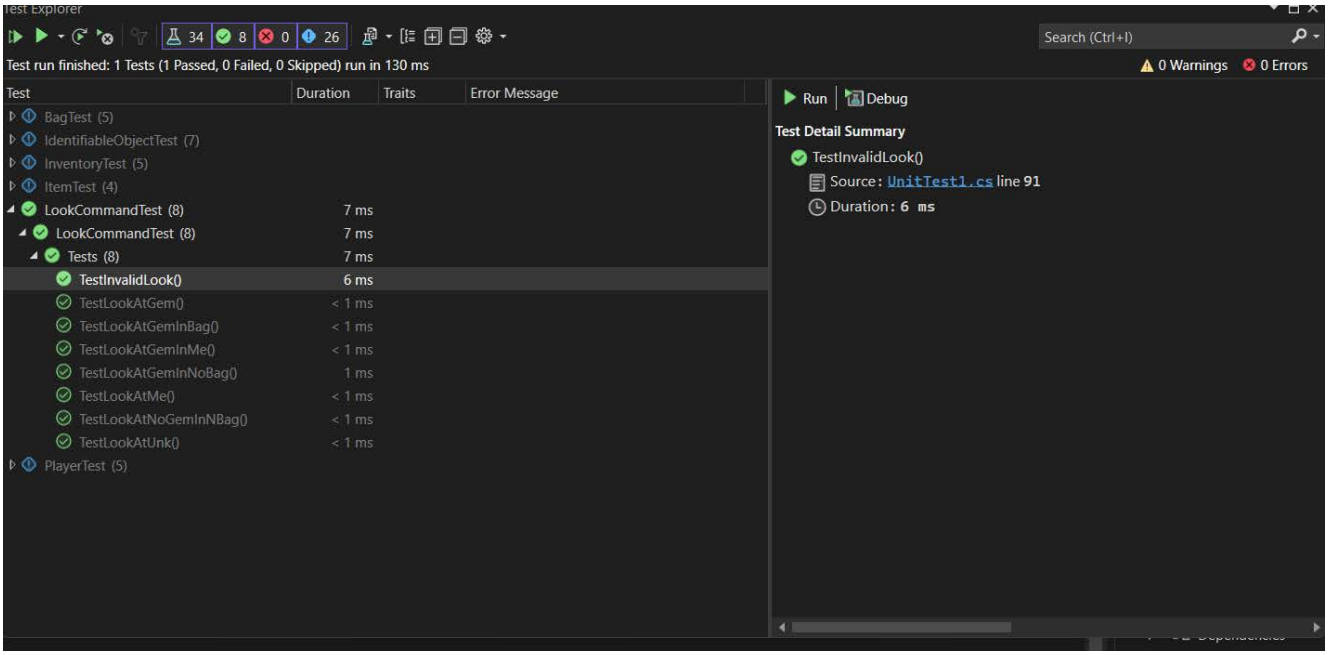
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace _6._1P
{
    public class LookCommand : Command
    {
        public LookCommand () : base(new string[] {"look"})
        {
        }

        public override string Execute(Player p, string[] text)
        {
            if (text.Length != 3 && text.Length != 5)
            {
                return "I don't know how to look like that";
            }

            if (text[0] != "look")
            {
                return "Error in look input";
            }

            if (text[1] != "at")
            {
                return "What do you want to look at?";
            }

            if (text.Length == 5 && text[3] != "in")
            {
                return "What do you want to look in?";
            }

            IHaveInventory container;
            if (text.Length == 3)
            {

                container = FetchContainer(p, "inventory");
            }

            else
            {

                container = FetchContainer(p, text[4]);
                if (container == null)
                {
                    return $"I cannot find the {text[4]}";
                }
            }

```

```
54              // Step 7: The item id is the 3rd word
55              string itemId = text[2];
56              return LookAtIn(itemId, container);
57
58          }
59          public IHaveInventory FetchContainer(Player p, string
              containerId)
60          {
61              if (containerId.ToLower() == "inventory")
62              {
63                  return p;
64              }
65
66              GameObject obj = p.Locate(containerId);
67              if (obj is IHaveInventory)
68              {
69                  return (IHaveInventory)obj; //container is bag(?)
70              }
71              return null;
72          }
73
74          public string LookAtIn(string itemId, IHaveInventory container)
75          {
76
77              GameObject item = container.Locate(itemId);
78              if (item == null)
79              {
80                  return $"I cannot find the {itemId} in
                      {container.Name}";
81              }
82              ////
83
84
85              return item.FullDescription;
86          }
87      }
88  }
89
```

```
 1  using _9._2C;
 2  using System;
 3  using System.Collections.Generic;
 4  using System.Linq;
 5  using System.Threading.Tasks;
 6  using NUnit.Framework;
 7  using System.Numerics;
 8
 9  namespace LookCommandTest
10  {
11      [TestFixture]
12      public class Tests
13      {
14          private Bag _bag;
15          private Item _item;
16          private Player _player;
17
18          [SetUp]
19          public void Setup()
20          {
21              _bag = new Bag(new string[] { "backpack" }, "Backpack",
                  "very gud backpack");
22              _item = new Item(new string[] { "diamond" }, "Diamond",
                  "shiny diamond");
23              _player = new Player("Nevan", "a desparate programmer");
24          }
25
26          [TestCase]
27          public void TestLookAtMe()
28          {
29              LookCommand look = new LookCommand();
30              string expectedDescription = look.Execute(_player, new
                  string[] { "look", "at", "inventory" });
31              Assert.AreEqual(_player.FullDescription,
                  expectedDescription);
32          }
33
34          [TestCase]
35          public void TestLookAtGem()
36          {
37              LookCommand look = new LookCommand();
38              _player.Inventory.Put(_item);
39              string result = look.Execute(_player, new string[]
                  { "look", "at", "diamond" });
40              Assert.AreEqual(_item.FullDescription, result);
41          }
42
43          [TestCase]
44          public void TestLookAtUnk()
45          {
46              LookCommand look = new LookCommand();
47              string result = look.Execute(_player, new string[]
                  { "look", "at", "diamond" });
```

```
48              Assert.AreEqual("I cannot find the diamond in Nevan",
                  result);
49          }
50
51          [TestCase]
52          public void TestLookAtGemInMe()
53          {
54              _player.Inventory.Put(_item);
55              LookCommand look = new LookCommand();
56              string expectedDescription = look.Execute(_player, new
                  string[] { "look", "at", "diamond", "in", "inventory" });
57              Assert.AreEqual(_item.FullDescription,
                  expectedDescription);
58          }
59
60          [TestCase]
61          public void TestLookAtGemInBag()
62          {
63              _player.Inventory.Put(_bag);
64              _bag.Inventory.Put(_item);
65              LookCommand look = new LookCommand();
66              string expectedDescription = look.Execute(_player, new
                  string[] { "look", "at", "diamond", "in", "backpack" });
67              Assert.AreEqual(_item.FullDescription,
                  expectedDescription);
68          }
69
70          [TestCase]
71          public void TestLookAtGemInNoBag()
72          {
73              //_player.Inventory.Put(_bag);
74              _bag.Inventory.Put(_item);
75              LookCommand look = new LookCommand();
76              string expectedDescription = look.Execute(_player, new
                  string[] { "look", "at", "diamond", "in", "backpack" });
77              Assert.AreEqual("I cannot find the backpack",
                  expectedDescription);
78          }
79
80          [TestCase]
81          public void TestLookAtNoGemInNBag()
82          {
83              _player.Inventory.Put(_bag);
84              //_bag.Inventory.Put(_item);
85              LookCommand look = new LookCommand();
86              string expectedDescription = look.Execute(_player, new
                  string[] { "look", "at", "diamond", "in", "backpack" });
87              Assert.AreEqual("I cannot find the diamond in Backpack",
                  expectedDescription);
88          }
89
90          [TestCase]
91          public void TestInvalidLook() //
```

```
92          {
93              LookCommand look = new LookCommand();
94              string expectedDescription = look.Execute(_player, new
                  string[] { "look", "around" });
95              Assert.AreEqual("I don't know how to look like that",
                  expectedDescription);
96
97              expectedDescription = look.Execute(_player, new string[]
                  { "hello", "104772183"});
98              Assert.AreEqual("I don't know how to look like that",
                  expectedDescription);
99
100             expectedDescription = look.Execute(_player, new string[]
                  { "look", "at", "Nguyen"});
101             Assert.AreEqual("I cannot find the Nguyen in Nevan",
                  expectedDescription);
102
103
104
105         }
106
107     }
108 }
```



```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace _6._1P
8  {
9      public class Player : GameObject, IHaveInventory
10     {
11         private Inventory _inventory = new Inventory();
12
13         public Player(string name, string description) : base(new string
             [] { "me", "inventory" }, name, description) { } //name and
             des gotten from GameObject
14         //help the class identify itself and its item, 3 batteries, 2
             from GO and 1 from IO
15         public GameObject Locate(string id)
16         {
17             if (AreYou(id))
18             {
19                 return this; //return then player object itself
20             }
21             return _inventory.Fetch(id);
22             //searches the inventory for an item with the given
                 identifier and returns it if found. If no item matches, it
                 returns null.
23         }
24
25         public override string FullDescription
26         {
27             get
28             {
29                 return $"You are {Name}, {base.FullDescription}\nYou are
                       carrying:\n{_inventory.ItemList}";
30             }
31         }
32
33         public Inventory Inventory { get { return _inventory; } }
34
35     }
36 }
37
```