```
 1  using Iteration4;
 2  using NUnit.Framework;
 3
 4  namespace UTLookCommand
 5  {
 6      public class Tests
 7      {
 8          private Player _player;
 9          private Item gem, spade;
10          private Bag bag;
11          private LookCommand lookCommand;
12
13          [SetUp]
14          public void Setup()
15          {
16              _player = new Player("Player1", "first player");
17              gem = new Item(new string[] { "gem", "a gem" }, "purple gem",
                  "big purple gem");
18              spade = new Item(new string[] { "spade" }, "purple spade",
                  "big purple spade");
19
20              lookCommand = new LookCommand();
21              bag = new Bag(new string[] { "testBag" }, "a bag", "contains
                  items");
22          }
23
24          [Test]
25          public void LookAtMeTest()
26          {
27              _player.Inventory.Put(gem);
28              var expectedOutcome = _player.FullDescription;
29              string desc = lookCommand.Execute(_player, new string[]
                  { "look", "at", "inventory" });
30              Assert.That(desc, Is.EqualTo(expectedOutcome));
31          }
32
33          [Test]
34          public void LookAtGemTest()
35          {
36              _player.Inventory.Put(gem);
37              var expectedOutcome = gem.FullDescription;
38              var result = lookCommand.Execute(_player, new string[]
                  { "look", "at", "gem" });
39              Assert.That(result, Is.EqualTo(expectedOutcome));
40          }
41
42          [Test]
43          public void LookAtUnkTest()
44          {
```

```
45              var expectedOutcome = "I can't find the gem";
46              var result = lookCommand.Execute(_player, new string[]
                  { "look", "at", "gem" });
47              Assert.That(result, Is.EqualTo(expectedOutcome));
48          }
49
50          [Test]
51          public void LookAtGemInMeTest()
52          {
53              _player.Inventory.Put(gem);
54              var expectedOutcome = gem.FullDescription;
55              var result = lookCommand.Execute(_player, new string[]
                  { "look", "at", "gem", "in", "me" });
56              Assert.That(result, Is.EqualTo(expectedOutcome));
57          }
58
59          [Test]
60          public void LookAtGeminBagTest()
61          {
62              _player.Inventory.Put(gem);
63              var expectedOutcome = gem.FullDescription;
64              var result = lookCommand.Execute(_player, new string[]
                  { "look", "at", "gem", "in", "Inventory" });
65              Assert.That(result, Is.EqualTo(expectedOutcome));
66          }
67
68          [Test]
69          public void LookAtGemInNoBagTest()
70          {
71              var expectedOutcome = "I can't find the bag";
72              var result = lookCommand.Execute(_player, new string[]
                  { "look", "at", "bag", "in", "me" });
73              Assert.That(result, Is.EqualTo(expectedOutcome));
74          }
75
76          [Test]
77          public void LookAtNoGemInBagTest()
78          {
79              var expectedOutcome = "I can't find the gem";
80              var result = lookCommand.Execute(_player, new string[]
                  { "look", "at", "gem", "in", "me" });
81              Assert.That(result, Is.EqualTo(expectedOutcome));
82          }
83
84          [Test]
85          public void InvalidLookTest()
86          {
87              var result0 = lookCommand.Execute(_player, new string[]
                  { "look", "there" });
```

```
88              Assert.That(result0, Is.EqualTo("I don't know how to look like
                   that"));
89
90              var result1 = lookCommand.Execute(_player, new string[]
                  { "there", "it", "is" });
91              Assert.That(result1, Is.EqualTo("Error in look input"));
92
93              var result2 = lookCommand.Execute(_player, new string[]
                  { "look", "over", "there" });
94              Assert.That(result2, Is.EqualTo("What do you want to look
                  at?"));
95
96              var result3 = lookCommand.Execute(_player, new string[]
                  { "look", "at", "gem", "over", "there" });
97              Assert.That(result3, Is.EqualTo("What do you want to look
                  in?"));
98          }
99      }
100 }
```