```
1  using Iteration1;
2  using Path = Iteration1.Path;
3
4  namespace TestMoveCommand
5  {
6      public class Tests
7      {
8          private Player _player;
9          private Location _location;
10         private Item _sword;
11         private Item _ak47;
12         private Item _grenade;
13         private Path _northpath;
14         private MoveCommand _move;
15
16         [SetUp]
17         public void Setup()
18         {
19             _player = new Player("Chien", "A boy with high curiosity");
20             _location = new Location(new string[] { "military base" },
                 "military base", "large area");
21             _sword = new Item(new string[] { "sword", "melee" }, "sword",
                 "Short range weapon");
22             _ak47 = new Item(new string[] { "ak47" }, "ak47", "Long range
                 weapon");
23             _grenade = new Item(new string[] { "grenade" }, "grenade",
                 "Very high damage weapon!");
24             _location.Inventory.Put(_sword);
25             _location.Inventory.Put(_ak47);
26             _player.CurrentLocation = _location;
27             _northpath = new Path(new string[] { "north" }, "hospital",
                 "this is a hospital");
28             _location.AddPath(_northpath);
29             _move = new MoveCommand();
30         }
31
32         [Test]
33         public void TestPathMovesPlayer()
34         {
35             _northpath.Move(_player);
36             Assert.That(_player.CurrentLocation.Name, Is.EqualTo
                   (_northpath.Destination));
37             Assert.Pass();
38         }
39
40         [Test]
41         public void TestGetPathFromLocation()
42         {
43             string id = "north";
```

```
44              Assert.That(_location.Locate(id), Is.EqualTo(_northpath));
45              Assert.Pass();
46          }
47
48          [Test]
49          public void TestPlayerLeaveLocation()
50          {
51              Assert.That(_move.Execute(_player, new string[] { "move
                  north" }), Is.EqualTo($"You have moved to
                  {_northpath.Destination}\n"));
52              _player.CurrentLocation = _location;
53              Assert.That(_move.Execute(_player, new string[] { "leave
                  north" }), Is.EqualTo($"You have moved to
                  {_northpath.Destination}\n"));
54              _player.CurrentLocation = _location;
55              Assert.That(_move.Execute(_player, new string[] { "go
                  north" }), Is.EqualTo($"You have moved to
                  {_northpath.Destination}\n"));
56              _player.CurrentLocation = _location;
57              Assert.That(_move.Execute(_player, new string[] { "head
                  north" }), Is.EqualTo($"You have moved to
                  {_northpath.Destination}\n"));
58              Assert.Pass();
59          }
60
61          [Test]
62          public void TestWrongCommand()
63          {
64              Assert.That(_move.Execute(_player, new string[] { "run
                  north" }), Is.EqualTo($"Wrong command input!\n"));
65              Assert.That(_move.Execute(_player, new string[] { "go home" }),
                    Is.EqualTo($"Wrong command input!\n"));
66              Assert.That(_move.Execute(_player, new string[] { "move
                  south" }), Is.EqualTo("There is not that path in this
                  location!\n"));
67              Assert.Pass();
68          }
69
70          [Test]
71          public void TestNotMoveWithUnvalidPath()
72          {
73              _move.Execute(_player, new string[] { "go home" });
74              _move.Execute(_player, new string[] { "move south" });
75              Assert.That(_player.CurrentLocation, Is.EqualTo(_location));
76              Assert.Pass();
77          }
78      }
79  }
```