```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Runtime.InteropServices;
5  using System.Text;
6  using System.Threading.Tasks;
7  using SwinAdventure;
8
9  namespace SwinAdventureTest
10 {
11     public class MoveCommandTest
12     {
13         Player player;
14         Location mainhall;
15         Location library;
16         Location principalroom;
17         Paths north;
18         Paths east;
19         MoveCommand movecommand;
20         [SetUp]
21         public void SetUp()
22         {
23             player = new Player("Truong Ngoc Gia Hieu", "A brave Swinburne
                 warrior");
24             mainhall = new Location("Community Hall", "A large hall for
                 celebrating events");
25             library = new Location("State library", "An interesting
                 library where people work and study");
26             principalroom = new Location("Principla Room", "Room of
                 principle of a school");
27             north = new Paths(new string[] { "north" }, "First door", "The
                 way to the library", library);
28             east = new Paths(new string[] { "east" }, "Second door", "The
                 way to the Principle Room", principalroom);
29             player.Location = mainhall;
30             mainhall.AddPath(north);
31             mainhall.AddPath(east);
32             movecommand = new MoveCommand();
33         }
34         [Test]
35         public void MoveTestSucccessful()
36         {
37             //Player starts at mainhall
38             Assert.AreEqual(mainhall, player.Location, "Player should
                 start in mainhall");
39             //Move go to north
40             string result = movecommand.Execute(player, new string[]
                 { "move", "north" });
41             //Assert that player's location changed to the library
```

```
42              Assert.AreEqual(library, player.Location, "Player should have
                  moved to the library");
43          }
44          [Test]
45          public void TestMoveEastSuccessful()
46          {
47              Assert.AreEqual(mainhall, player.Location);
48              string result = movecommand.Execute(player, new string[]
                  { "go", "east" });
49              Assert.AreEqual(principalroom, player.Location);
50          }
51          [Test]
52          public void MoveTestFail()
53          {
54              Location startingLocation = player.Location;
55              string result = movecommand.Execute(player, new string[]
                  { "move", "south" });
56              Assert.AreEqual(startingLocation, player.Location, "Player
                  should still be in the main hall after failed move");
57          }
58      }
59  }
60
```