

SWINBURNE UNIVERSITY OF TECHNOLOGY

OBJECT ORIENTED PROGRAMMING

DOUBTFIRE SUBMISSION

Credit Task 4.2: Case Study Iteration 2

Submitted By:

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

Tutor:

Cliff WARREN



```
1  using System;
2  namespace SwinAdventure
3  {
4      public class GameObject:IdentifiableObject
5      {
6          private string _description;
7          private string _name;
8
9          public GameObject(string[] ids, string name, string desc): base (ids)
10         {
11             _name = name;
12             _description = desc;
13         }
14
15         public string ShortDescription
16         {
17             get
18             {
19                 return _name + " " + FirstId;
20             }
21         }
22
23         public virtual string FullDescription
24         {
25             get
26             {
27                 return _description;
28             }
29         }
30
31         public string Name
32         {
33             get
34             {
35                 return _name;
36             }
37
38             set
39             {
40                 _name = value;
41             }
42         }
43     }
44 }
```

```
1  using System;
2  namespace SwinAdventure
3  {
4      public class Player:GameObject
5      {
6          private Inventory _inventory;
7
8          public Player(string name,string desc): base(new string[] { "me",
9              ↪ "inventory"} , name, desc)
10         {
11             this.Name = name;
12
13             _inventory = new Inventory();
14         }
15
16         public GameObject Locate(string id)
17         {
18             if (this.AreYou(id))
19             {
20                 return this;
21             }
22             else if (this.Inventory.HasItem(id))
23             {
24                 return this.Inventory.Fetch(id);
25             }
26
27             return null;
28         }
29
30         /*GameObject result = null;
31         if (AreYou(id))
32         {
33             result = Inventory.Fetch(id);
34             return result;
35         }
36         else
37         {
38             return null;
39         }
40         */
41
42         public override string FullDescription
43         {
44             get
45             {
46                 return Inventory.ItemList;
47             }
48         }
49
50         public Inventory Inventory
51         {
52             get
53             {
```

```
53         return _inventory;
54     }
55 }
56 }
57 }
```

```
1  using System;
2  using NUnit.Framework;
3  namespace SwinAdventure
4  {
5      [TestFixture]
6      public class TestPlayer
7      {
8          Inventory tester = new Inventory();
9          Item shovel = new Item(new String[] { "shovel", "spade" },
10                                  "a shovel", "You are carrying:");
11          Player testing = new Player( "shovel", " a shovel");
12
13          [Test]
14          public void TestPlayerIdentifiable()
15          {
16              Assert.IsTrue(testing.AreYou("inventory"));
17          }
18
19          [Test]
20          public void TestPlayerLocateItem()
21          {
22              testing.Locate("shovel");
23              Assert.AreEqual("a shovel shovel\n", tester.ItemList);
24              Assert.IsTrue(testing.HasItem("shovel"));
25          }
26
27          [Test]
28          public void TestPlayerLocateItself()
29          {
30              Assert.IsTrue(testing.Locate("me") == testing);
31          }
32
33          [Test]
34          public void TestPlayerLocateNothing()
35          {
36              testing.AreYou("me");
37              Assert.IsNull(testing.Fetch("me"));
38          }
39
40          [Test]
41          public void TestPlayerFullDescription()
42          {
43              tester.Put(shovel);
44              StringAssert.Contains("You are carrying: \n" + "a shovel", "You are
45                                  ↪ carrying: \n" + "a shovel" );
46          }
47      }
```

```
1  using System;
2  namespace SwinAdventure
3  {
4      public class Item:GameObject
5      {
6          public Item(string[] idents, string name, string desc):base(idents, name,
              ↳ desc)
7          {
8          }
9      }
10 }
```

```
1  using System;
2  using NUnit.Framework;
3  namespace SwinAdventure
4  {
5      [TestFixture]
6      public class TestItem
7      {
8          Item shovel = new Item(new String[] { "shovel", "spade" },
9                                  "a shovel", "This is a might fine ...");
10
11         [Test]
12         public void Testidentify()
13         {
14             Assert.True(shovel.AreYou("sPade"));
15         }
16
17         [Test]
18         public void Testshortdescription()
19         {
20             StringAssert.Contains("a shovel shovel", shovel.ShortDescription);
21         }
22
23         [Test]
24         public void Testfulldescription()
25         {
26             StringAssert.Contains("This is a might fine ...",
27                                   ⇨ shovel.FullDescription);
28         }
29     }
30 }
```

```
1  using System;
2  using System.Collections.Generic;
3  namespace SwinAdventure
4  {
5      public class Inventory
6      {
7          List<Item> _items = new List<Item>();
8
9          public Inventory()
10         {
11         }
12
13         public bool HasItem(string id)
14         {
15             foreach (Item itm in _items)
16             {
17                 if (itm.AreYou(id))
18                 {
19                     return true;
20                 }
21             }
22             return false;
23         }
24
25         public void Put(Item itm)
26         {
27             _items.Add(itm);
28         }
29
30         public Item Take(string id)
31         {
32             Item result = null;
33             foreach (Item itm in _items)
34             {
35                 if (itm.AreYou(id))
36                 {
37                     result = itm;
38                 }
39             }
40             _items.Remove(result);
41             return result;
42         }
43
44         public Item Fetch(string id)
45         {
46             foreach (Item itm in _items)
47             {
48                 if (itm.AreYou(id))
49                 {
50                     return itm;
51                 }
52             }
53             return null;
54         }
55     }
```



```
54     }
55
56     public string ItemList
57     {
58         get
59         {
60             string result = "";
61             foreach (Item itm in _items)
62             {
63                 result += itm.ShortDescription + "\n";
64             }
65             if (result == null)
66             {
67                 return "There are no items in the inventory";
68             }
69             return result;
70         }
71     }
72 }
73 }
```

```
1  using System;
2  using NUnit.Framework;
3  namespace SwinAdventure
4  {
5      [TestFixture]
6      public class TestInventory
7      {
8          Inventory tester = new Inventory();
9          Item shovel = new Item(new String[] { "shovel", "spade" },
10                                 "a shovel", "This is a might fine ...");
11
12          [Test]
13          public void Testfinditem()
14          {
15              tester.Put(shovel);
16              Assert.True(tester.HasItem("shovel"));
17          }
18
19          [Test]
20          public void Testnoitem()
21          {
22              tester.Put(shovel);
23              Assert.False(tester.HasItem("banana"));
24          }
25
26          [Test]
27          public void Testfetchitem()
28          {
29              tester.Put(shovel);
30              Assert.AreEqual(shovel, tester.Fetch("shovel"));
31              Assert.IsTrue(tester.HasItem("shovel"));
32          }
33
34          [Test]
35          public void Testtakeitem()
36          {
37              tester.Put(shovel);
38              Assert.AreEqual(shovel, tester.Take("shovel"));
39              Assert.IsTrue(tester.HasItem("shovel"));
40          }
41
42          [Test]
43          public void TestItemlist()
44          {
45              tester.Take("shovel");
46              Assert.AreEqual("a shovel shovel" + "\n", tester.ItemList);
47          }
48      }
```

