

SWINBURNE UNIVERSITY OF TECHNOLOGY

OBJECT ORIENTED PROGRAMMING (2022 S1)

DOUBTFIRE SUBMISSION

---

## Task 5.1P: Case Study Iteration 3: Bags

---

*Submitted By:*

Vaissheenavi PRABAKARAN

103508183

2022/04/25 23:06

*Tutor:*

Jai CORNES

April 26, 2022



```
1  using System;
2  using System.Collections.Generic;
3
4  namespace Task_5._1
5  {
6      public class IdentifiableObject
7      {
8          private List<string> _identifiers = new List<string>();
9
10
11
12         public IdentifiableObject(string[] idents)
13         {
14             foreach (string id in idents)
15             {
16                 _identifiers.Add(id.ToLower());
17             }
18         }
19
20         //private List<string> _identifiers= new List<string>();
21
22
23
24
25         public bool AreYou(string id)
26         {
27             //return _identifiers.Contains(id.ToLower());
28
29             foreach (string idAY in _identifiers)
30             {
31                 if (id.ToLower() == idAY)
32                 {
33                     return true;
34                 }
35
36                 //return false;
37             }
38
39             return false;
40         }
41
42
43         public string FirstID
44         {
45             get
46             {
47                 if (_identifiers.Count > 0)
48                 {
49                     return _identifiers[0];
50                 }
51
52                 return "";
53             }
54         }
55     }
```

```
54         }
55
56         public void AddIdentifier(string id)
57         {
58             //id = id.ToLower();
59             _identifiers.Add(id.ToLower());
60
61             return;
62         }
63     }
64 }
```

```
1  using System;
2
3  namespace Task_5._1
4  {
5      public abstract class GameObject : IdentifiableObject
6      {
7          private string _description;
8          private string _name;
9
10
11
12
13         public string Name
14         {
15             get
16             {
17                 return _name;
18             }
19         }
20
21
22         public string ShortDescription
23         {
24             get
25             {
26                 return _name + " (" + FirstID + ")";
27             }
28         }
29
30
31
32         public virtual string FullDescription
33         {
34             get
35             {
36                 return _description;
37             }
38         }
39
40
41
42         public GameObject(string[] ids, string name, string desc) : base(ids)
43         {
44             _name = name;
45             _description = desc;
46         }
47
48     }
49 }
50
```

```
1  using System;
2  namespace Task_5._1
3  {
4      public class Player : GameObject
5      {
6          //already in GameObject
7          //private string _description;
8          //private string _name;
9          private Inventory inventory;
10
11
12         public Player(string name, string desc) : base(new string[] { "myself",
13             ↪ "inventory" }, name, desc)
14         {
15             inventory = new Inventory();
16         }
17
18         public GameObject Locate(string id)
19         {
20             if (AreYou(id))
21             {
22                 return this;
23             }
24
25             else if (inventory.HasItem(id))
26             {
27                 return inventory.Fetch(id);
28             }
29
30             return null;
31         }
32
33         public override string FullDescription
34         {
35             get
36             {
37                 return inventory.ItemList;
38                 //string inventorydescription = "You're carrying: " +
39                 ↪ Inventory.ItemList;
40                 //return inventorydescription;
41             }
42         }
43
44         public Inventory Inventory
45         {
46             get
47             {
48                 return inventory;
49             }
50         }
51     }
```

```
1  using System;
2  using System.Collections.Generic;
3  namespace Task_5._1
4  {
5      public class Inventory
6      {
7          private List<Item> _items = new List<Item>();
8
9
10         public Inventory()
11         {
12             //foreach (Item i in _items)
13             //{
14                 //if (i.AreYou(id))
15                 //{
16                     //return true;
17                 //}
18
19                 //else
20                 //{
21                     //return false;
22                 //}
23
24             //}
25
26
27             //_items = new List<Item>();
28
29         }
30
31         public bool HasItem(string id)
32         {
33             foreach (Item i in _items)
34             {
35                 if (i.AreYou(id))
36                 {
37                     return true;
38                 }
39
40             }
41             return false;
42
43         }
44
45
46
47         public void Put(Item itm)
48         {
49             _items.Add(itm);
50
51         }
52
53
```

```
54     public Item Fetch(string id)
55     {
56         foreach (Item i in _items)
57         {
58             if (i.AreYou(id))
59             {
60                 Item itemToFetch = i;
61
62                 //return true;
63                 return itemToFetch;
64             }
65
66             //return null;
67
68         }
69
70         return null;
71     }
72
73
74     public Item Take(string id)
75     {
76         Item i = Fetch(id);
77         if (i != null)
78         {
79             _items.Remove(i);
80             return i;
81         }
82
83         return null;
84     }
85
86
87     public string ItemList
88     {
89         get
90         {
91             string iList = "";
92             foreach (Item i in _items)
93             {
94                 iList += "\t" + i.ShortDescription + "\n";
95             }
96
97             if (iList == null)
98             {
99                 return "Item not found!";
100             }
101
102             return iList;
103         }
104     }
105 }
106 }
```

107 }



```
1  using System;
2  namespace Task_5._1
3  {
4      public class Item : GameObject
5      {
6          //already in GameObject
7          //private string _description;
8          //private string _name;
9
10
11         public Item(string[] idents, string name, string desc) : base(idents, name,
            ↪ desc)
12         {
13             //_name = name;
14             //_description = desc;
15         }
16     }
17 }
```

```
1  using System;
2  using System.Collections.Generic;
3  namespace Task_5._1
4  {
5      public class Bag : Item
6      {
7          private Inventory _inventory;
8
9          public Bag(string[] ids, string name, string desc) : base(ids, name, desc)
10         {
11             _inventory = new Inventory();
12         }
13
14         public Inventory Inventory
15         {
16             get
17             {
18                 return _inventory;
19             }
20         }
21
22         public override string FullDescription
23         {
24             get
25             {
26                 return "\tIn " + Name + " you can see : " + _inventory.ItemList +
27                     ↵ "\n";
28             }
29         }
30
31         public GameObject Locate(string id)
32         {
33             if (AreYou(id))
34             {
35                 return this;
36             }
37
38             else if (_inventory.HasItem(id))
39             {
40                 return _inventory.Fetch(id);
41             }
42
43             return null;
44         }
45     }
46 }
47 }
```

```
1  using NUnit.Framework;
2
3  namespace Task_5._1
4  {
5      public class BagUnitTests
6      {
7
8          public Bag _bag;
9          public Bag _bag2;
10         public Item _mirror;
11         public Item _hairBrush;
12
13
14         [SetUp]
15         public void Setup()
16         {
17             //_inventory = new Inventory();
18             _bag = new Bag (new string[] { "yellowBag", "yB"}, "a bag", "This is a
19                 ↪ yellow bag....");
20             _mirror = new Item(new string[] { "mirror" }, "a mirror", "This is a
21                 ↪ room item....");
22             _hairBrush = new Item(new string[] { "hairbrush" }, "a hairbrush",
23                 ↪ "This is a room item....");
24             _bag2 = new Bag(new string[] { "blueBag", "bB" }, "a bag", "This is a
25                 ↪ blue bag....");
26
27         }
28
29         [Test]
30         public void LocatesNothingTest()
31         {
32             Assert.IsNull(_bag.Locate("cap"));
33
34         }
35
36         [Test]
37         public void LocatesItemTest()
38         {
39             _bag.Inventory.Put(_mirror);
40
41             Assert.AreEqual(_mirror, _bag.Locate("mirror"));
42
43         }
44
45         [Test]
46         public void FullDescTest()
47         {
48             _bag.Inventory.Put(_mirror);
49             Assert.AreEqual("\tIn a bag you can see : \ta mirror (mirror)\n\n",
50                 ↪ _bag.FullDescription);
51         }
52     }
53 }
```

```
49         //_player.FullDescription, "You're carrying: " + _inventory.ItemList);
50
51     }
52
53     [Test]
54     public void LocatesItselfTest()
55     {
56         Assert.AreEqual(_bag, _bag.Locate("yellowBag"));
57
58     }
59
60
61     [Test]
62     public void BagInBagTest()
63     {
64         //Bag firstbag = _bag = new Bag(new string[] { "yellowBag", "yB" }, "a
        ↪ bag", "This is a yellow bag....");
65         //Bag secondbag = new Bag(new string[] { "blueBag", "bB" }, "a bag",
        ↪ "This is a blue bag....");
66
67         _bag.Inventory.Put(_bag2);
68
69         Assert.AreEqual(_bag2, _bag.Locate("blueBag"));
70
71         _bag.Inventory.Put(_hairBrush);
72
73         Assert.AreEqual(_hairBrush, _bag.Locate("hairbrush"));
74
75         _bag2.Inventory.Put(_mirror);
76
77         Assert.AreNotEqual(_mirror, _bag.Locate("_mirror"));
78     }
79 }
80 }
```

