

Selected files

9 printable files

Week_4/4.2/SwinAdventure/GameObject.cs
Week_4/4.2/SwinAdventure/IdentifiableObject.cs
Week_4/4.2/SwinAdventure/Inventory.cs
Week_4/4.2/SwinAdventure/Item.cs
Week_4/4.2/SwinAdventure/Player.cs
Week_4/4.2/SwinAdventure/Program.cs
Week_4/4.2/TestInventory/TestInventory.cs
Week_4/4.2/TestItem/TestItem.cs
Week_4/4.2/TestPlayer/TestPlayer.cs

Week_4/4.2/SwinAdventure/GameObject.cs

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5
6  namespace SwinAdventure
7  {
8      public abstract class GameObject : IdentifiableObject
9      {
10         private string _description, _name;
11
12         public GameObject(string[] idents, string name, string desc) : base(idents)
13         {
14             _name = name;
15             _description = desc;
16         }
17
18         public string Name => _name;
19
20         public string ShortDescription => $"{Name} ({FirstId})";
21
22         public virtual string FullDescription => _description;
23     }
24 }
```

Week_4/4.2/SwinAdventure/IdentifiableObject.cs

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5
6  namespace SwinAdventure
7  {
8
9      public class IdentifiableObject
10     {
```

```

11     private List<string> _identifiers = new List<string>();
12
13     public IdentifiableObject(string[] idents)
14     {
15         foreach (string id in idents)
16         {
17             AddIdentifier(id);
18         }
19     }
20
21     public bool AreYou(string id)
22     {
23         return _identifiers.Contains(id.ToLower());
24     }
25
26     public string FirstId
27     {
28         get
29         {
30             if (_identifiers.Count > 0)
31             {
32                 return _identifiers[0];
33             }
34
35             return "";
36         }
37     }
38
39     public void AddIdentifier(string id)
40     {
41         _identifiers.Add(id.ToLower());
42     }
43
44     public void PrivilegeEscalation(string pin)
45     {
46         if (pin != "4794")
47             return;
48
49         if (_identifiers.Count == 0)
50         {
51             AddIdentifier("104844794");
52         }
53         else
54         {
55             _identifiers[0] = "104844794";
56         }
57     }
58 }
59 }

```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5
6 namespace SwinAdventure
7 {
8     public class Inventory : GameObject
9     {
10         private List<Item> _items;
11
12         public Inventory() : base(new string[] { "inventory" }, "inventory", "The
player's inventory")
13         {
14             _items = new List<Item>();
15         }
16
17         public string ItemList
18         {
19             get
20             {
21                 List<string> itemsDesc = new List<string>();
22                 foreach (Item item in _items)
23                 {
24                     itemsDesc.Add("\t" + item.ShortDescription);
25                 }
26                 return string.Join("\n", itemsDesc);
27             }
28         }
29
30         public bool HasItem(string id)
31         {
32             foreach (Item item in _items)
33             {
34                 if (item.AreYou(id))
35                 {
36                     return true;
37                 }
38             }
39             return false;
40         }
41
42         public void Put(Item itm)
43         {
44             _items.Add(itm);
45         }
46
47         public Item? Take(string id)
48         {
49             foreach (Item item in _items)
50             {
```

```

51         if (item.AreYou(id))
52         {
53             _items.Remove(item);
54             return item;
55         }
56     }
57     return null;
58 }
59
60 public Item? Fetch(string id)
61 {
62     foreach (Item item in _items)
63     {
64         if (item.AreYou(id))
65         {
66             return item;
67         }
68     }
69     return null;
70 }
71 }
72 }

```

Week_4/4.2/SwinAdventure/Item.cs

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5
6  namespace SwinAdventure
7  {
8      public class Item : GameObject
9      {
10         public Item(string[] idents, string name, string desc) : base(idents, name,
11         desc)
12         {
13         }
14     }
15 }

```

Week_4/4.2/SwinAdventure/Player.cs

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5
6  namespace SwinAdventure
7  {
8      public class Player : GameObject
9      {
10         private Inventory _inventory;

```

```

11
12     public Player(string name, string desc) : base(new string[] { "me",
"inventory" }, name, desc)
13     {
14         _inventory = new Inventory();
15     }
16
17     public GameObject? Locate(string id)
18     {
19         if (AreYou(id))
20             return this;
21
22         if (_inventory.HasItem(id))
23             return _inventory.Fetch(id);
24
25         return null;
26     }
27
28     public override string FullDescription
29     {
30         get
31         {
32             return $"You are {Name}, {base.FullDescription}\n" +
33                 $"You are carrying:\n{_inventory.ItemList}";
34         }
35     }
36
37     public Inventory Inventory => _inventory;
38 }
39 }

```

Week_4/4.2/SwinAdventure/Program.cs

```

1 namespace SwinAdventure
2 {
3     class Program
4     {
5         static void Main()
6         {
7             string expected = "\ta sharp sword (sword)\n" +
8                             "\ta blunt axe (axe)";
9             System.Console.WriteLine(expected);
10        }
11    }
12 }

```

Week_4/4.2/TestInventory/TestInventory.cs

```

1 using NuGet.Frameworks;
2 using SwinAdventure;
3
4 namespace TestInventory

```

```
5 {
6     public class Tests
7     {
8         private Inventory _inventory;
9
10        [SetUp]
11        public void Setup()
12        {
13            _inventory = new Inventory();
14            _inventory.Put(new Item(new string[] { "sword" }, "a sharp sword", "A
sharp sword for cutting things"));
15            _inventory.Put(new Item(new string[] { "axe" }, "a blunt axe", "A blunt
axe for bludgeoning things"));
16        }
17
18        [Test]
19        public void TestFindItem()
20        {
21            Assert.IsTrue(_inventory.HasItem("sword"));
22            Assert.IsTrue(_inventory.HasItem("axe"));
23        }
24
25        [Test]
26        public void TestNoItemFind()
27        {
28            Assert.IsFalse(_inventory.HasItem("dagger"));
29        }
30
31        [Test]
32        public void TestFetchItem()
33        {
34            Item item = _inventory.Fetch("sword");
35            Assert.IsTrue(item.AreYou("sword"));
36            Assert.IsTrue(_inventory.HasItem("sword"));
37        }
38
39        [Test]
40        public void TestTakeItem()
41        {
42            Item item = _inventory.Take("axe");
43            Assert.IsTrue(item.AreYou("axe"));
44            Assert.IsFalse(_inventory.HasItem("axe"));
45        }
46
47        [Test]
48        public void TestItemList()
49        {
50            string expected = "\ta sharp sword (sword)\n\taxe (axe)";
51            Assert.AreEqual(expected, _inventory.ItemList);
52        }
53    }
```

```
54 | }
```

Week_4/4.2/TestItem/TestItem.cs

```
1  using SwinAdventure;
2
3  namespace TestItem
4  {
5      public class Tests
6      {
7          private Item _item;
8          [SetUp]
9          public void Setup()
10         {
11             _item = new Item(new string[] { "sword" }, "a sharp sword", "A sharp
sword for cutting things");
12         }
13
14         [Test]
15         public void TestItemIsIdentifiable()
16         {
17             Assert.IsTrue(_item.AreYou("sword"));
18             Assert.IsFalse(_item.AreYou("axe"));
19         }
20
21         [Test]
22         public void TestShortDescription()
23         {
24             Assert.AreEqual("a sharp sword (sword)", _item.ShortDescription);
25         }
26
27         [Test]
28         public void TestFullDescription()
29         {
30             Assert.AreEqual("A sharp sword for cutting things",
_item.FullDescription);
31         }
32
33         [Test]
34         public void TestPrivilegeEscalation()
35         {
36             _item.PrivilegeEscalation("4794");
37             Assert.AreEqual("104844794", _item.FirstId);
38         }
39     }
40 }
```

Week_4/4.2/TestPlayer/TestPlayer.cs

```
1  using SwinAdventure;
2
3  namespace TestPlayer
4  {
```

```
5 public class Tests
6 {
7     private Player _player;
8
9     [SetUp]
10    public void Setup()
11    {
12        _player = new Player("Minh An", "an AI enthusiast");
13        _player.Inventory.Put(new Item(new string[] { "sword" }, "a sharp sword",
14    "A sharp sword for cutting things"));
15        _player.Inventory.Put(new Item(new string[] { "axe" }, "a blunt axe", "A
16    blunt axe for bludgeoning things"));
17    }
18
19    [Test]
20    public void TestPlayerIdentifiable()
21    {
22        Assert.IsTrue(_player.AreYou("me"));
23        Assert.IsTrue(_player.AreYou("inventory"));
24    }
25
26    [Test]
27    public void TestPlayerLocatesItems()
28    {
29        Assert.IsTrue(_player.Locate("sword")!.AreYou("sword"));
30        Assert.IsTrue(_player.Locate("axe")!.AreYou("axe"));
31    }
32
33    [Test]
34    public void TestPlayerLocatesItself()
35    {
36        Assert.IsTrue(_player.Locate("me")!.AreYou("me"));
37        Assert.IsTrue(_player.Locate("inventory")!.AreYou("inventory"));
38    }
39
40    [Test]
41    public void TestPlayerLocatesNothing()
42    {
43        Assert.IsNull(_player.Locate("dagger"));
44    }
45
46    [Test]
47    public void TestPlayerFullDescription()
48    {
49        string expected = "You are Minh An, an AI enthusiast\n" +
50            "You are carrying:\n" +
51            "\ta sharp sword (sword)\n" +
52            "\ta blunt axe (axe)";
53        Assert.AreEqual(expected, _player.FullDescription);
54    }
55 }
```


Screenshot of test output:

