Selected files

8 printable files

Week_5/5.2/SwinAdventure/Bag.cs
Week_5/5.2/SwinAdventure/GameObject.cs
Week_5/5.2/SwinAdventure/IdentifiableObject.cs
Week_5/5.2/SwinAdventure/Inventory.cs
Week_5/5.2/SwinAdventure/Item.cs
Week_5/5.2/SwinAdventure/Player.cs
Week_5/5.2/SwinAdventure/Program.cs
Week_5/5.2/TestBag/TestBag.cs

Week_5/5.2/SwinAdventure/Bag.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography;
using System.Threading.Tasks;

namespace SwinAdventure
{
    public class Bag : Item
    {
        private Inventory _inventory;
        public Bag(string[] idents, string name, string desc) : base(idents, name,
desc)
        {
            _inventory = new Inventory();
        }

        public GameObject? Locate(string id)
        {
            if (AreYou(id))
                return this;

            if (_inventory.HasItem(id))
                return _inventory.Fetch(id);

            return null;
        }

        public Inventory Inventory => _inventory;
        public override string FullDescription
        {
            get
            {
                return $"In the {Name} you can see:\n{_inventory.ItemList}";
            }
        }
    }
```

```
}
```

Week_5/5.2/SwinAdventure/GameObject.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace SwinAdventure
{
    public abstract class GameObject : IdentifiableObject
    {
        private string _description, _name;

        public GameObject(string[] idents, string name, string desc) : base(idents)
        {
            _name = name;
            _description = desc;
        }

        public string Name => _name;

        public string ShortDescription => $"{Name} ({FirstId})";

        public virtual string FullDescription => _description;
    }
}
```

Week_5/5.2/SwinAdventure/IdentifiableObject.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace SwinAdventure
{

    public class IdentifiableObject
    {
        private List<string> _identifiers = new List<string>();

        public IdentifiableObject(string[] idents)
        {
            foreach (string id in idents)
            {
                AddIdentifier(id);
            }
        }

        public bool AreYou(string id)
```

```
        {
            return _identifiers.Contains(id.ToLower());
        }

        public string FirstId
        {
            get
            {
                if (_identifiers.Count > 0)
                {
                    return _identifiers[0];
                }

                return "";
            }
        }

        public void AddIdentifier(string id)
        {
            _identifiers.Add(id.ToLower());
        }

        public void PrivilegeEscalation(string pin)
        {
            if (pin != "4794")
                return;

            if (_identifiers.Count == 0)
            {
                AddIdentifier("12");
            }
            else
            {
                _identifiers[0] = "12";
            }
        }
    }
}
```

Week_5/5.2/SwinAdventure/Inventory.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace SwinAdventure
{
    public class Inventory : GameObject
    {
        private List<Item> _items;

```

```
        public Inventory() : base(new string[] { "inventory" }, "inventory", "The player's inventory")
        {
            _items = new List<Item>();
        }

        public string ItemList
        {
            get
            {
                List<string> itemsDesc = new List<string>();
                foreach (Item item in _items)
                {
                    itemsDesc.Add("\t" + item.ShortDescription);
                }
                return string.Join("\n", itemsDesc);
            }
        }

        public bool HasItem(string id)
        {
            foreach (Item item in _items)
            {
                if (item.AreYou(id))
                {
                    return true;
                }
            }
            return false;
        }

        public void Put(Item itm)
        {
            _items.Add(itm);
        }

        public Item? Take(string id)
        {
            foreach (Item item in _items)
            {
                if (item.AreYou(id))
                {
                    _items.Remove(item);
                    return item;
                }
            }
            return null;
        }

        public Item? Fetch(string id)
        {
```

```
            foreach (Item item in _items)
            {
                if (item.AreYou(id))
                {
                    return item;
                }
            }
            return null;
        }
    }
}
```

Week_5/5.2/SwinAdventure/Item.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace SwinAdventure
{
    public class Item : GameObject
    {
        public Item(string[] idents, string name, string desc) : base(idents, name,
desc)
        {
        }
    }
}
```

Week_5/5.2/SwinAdventure/Player.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace SwinAdventure
{
    public class Player : GameObject
    {
        private Inventory _inventory;

        public Player(string name, string desc) : base(new string[] { "me",
"inventory" }, name, desc)
        {
            _inventory = new Inventory();
        }

        public GameObject? Locate(string id)
        {
            if (AreYou(id))
                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_5/5.2/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_5/5.2/TestBag/TestBag.cs

```
1   namespace TestBag;
2   using SwinAdventure;
3
4   public class Tests
5   {
6       private Bag _b1;
7       private Bag _b2;
8       private Item _item1;
9       private Item _item2;
10
11      [SetUp]
12      public void Setup()
13      {
14          _b1 = new Bag(new string[] { "bag1" }, "Bag 1", "A huge bag");
15          _b2 = new Bag(new string[] { "bag2" }, "Bag 2", "A small bag");
```

```
16
17          _item1 = new Item(new string[] { "shovel" }, "Shovel", "A shovel");
18          _b1.Inventory.Put(_item1);
19
20          _item2 = new Item(new string[] { "sword" }, "Sword", "A sword");
21          _b2.Inventory.Put(_item2);
22
23          _b1.Inventory.Put(_b2);
24      }
25
26      [Test]
27      public void TestBagLocateItem()
28      {
29          Assert.AreEqual(_item1, _b1.Locate("shovel"));
30          Assert.IsTrue(_b1.Inventory.HasItem("shovel"));
31      }
32
33      [Test]
34      public void TestBagLocateItself()
35      {
36          Assert.AreEqual(_b1, _b1.Locate("bag1"));
37      }
38
39      [Test]
40      public void TestBagLocateNothing()
41      {
42          Assert.IsNull(_b1.Locate("axe"));
43      }
44
45      [Test]
46      public void TestBagFullDescription()
47      {
48          Assert.AreEqual("In the Bag 1 you can see:\n\tShovel (shovel)\n\tBag 2
    (bag2)", _b1.FullDescription);
49      }
50
51      [Test]
52      public void TestBagInBag()
53      {
54          Assert.AreEqual(_b2, _b1.Locate("bag2"));
55          Assert.AreEqual(_item1, _b1.Locate("shovel"));
56          Assert.IsNull(_b1.Locate("sword"));
57      }
58
59      [Test]
60      public void TestBagHasPriviledgeItem()
61      {
62          Item item = new Item(new string[] { "axe" }, "Axe", "An axe");
63          item.PrivilegeEscalation("4794");
64          _b2.Inventory.Put(item);
65          Assert.IsNull(_b1.Locate("12"));
```

```
    }
}

```

Screenshot of test output: