

Selected files

12 printable files

Week_7\7.2\SwinAdventure\Bag.cs
Week_7\7.2\SwinAdventure\Command.cs
Week_7\7.2\SwinAdventure\GameObject.cs
Week_7\7.2\SwinAdventure\IdentifiableObject.cs
Week_7\7.2\SwinAdventure\IHaveInventory.cs
Week_7\7.2\SwinAdventure\Inventory.cs
Week_7\7.2\SwinAdventure\Item.cs
Week_7\7.2\SwinAdventure\Location.cs
Week_7\7.2\SwinAdventure\LookCommand.cs
Week_7\7.2\SwinAdventure\Player.cs
Week_7\7.2\SwinAdventure\Program.cs
Week_7\7.2\TestLocation\TestLocation.cs

Week_7\7.2\SwinAdventure\Bag.cs

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Security.Cryptography;
5  using System.Threading.Tasks;
6
7  namespace SwinAdventure
8  {
9      public class Bag : Item, IHaveInventory
10     {
11         private Inventory _inventory;
12         public Bag(string[] idents, string name, string desc) : base(idents, 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 Inventory Inventory => _inventory;
29         public override string FullDescription
30         {
31             get
32             {
33                 return $"In the {Name} you can see:\n{_inventory.ItemList}";
34             }
35         }
36     }
37 }
```

```
35     }
36 }
37 }
```

Week_7\7.2\SwinAdventure\Command.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 Command : IdentifiableObject
9      {
10         public Command(string[] ids) : base(ids)
11         {
12         }
13
14         public abstract string Execute(Player p, string[] text);
15     }
16 }
```

Week_7\7.2\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_7\7.2\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("12");
52     }
53     else
54     {
55         _identifiers[0] = "12";
56     }
57 }
58 }
59 }

```

Week_7\7.2\SwinAdventure\IHaveInventory.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 interface IHaveInventory
9      {
10         public GameObject? Locate(string id);
11         public string Name { get; }
12     }
13 }

```

Week_7\7.2\SwinAdventure\Inventory.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 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 }

```

```

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, desc)
11         {
12         }
13     }
14 }

```

Week_7\7.2\SwinAdventure\Location.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 Location : GameObject, IHaveInventory
9     {
10         private Inventory _inventory;
11
12         public Location(string name, string desc) : base(new string[] { "location" }, name,
13 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             return _inventory.Fetch(id);
26         }
27
28         public override string FullDescription
29         {
30             get
31             {
32                 return $"You are in the {Name}.\n{base.FullDescription}\nIn this location, you
can see:\n{_inventory.ItemList}";
33             }
34         }
35     }
36 }

```



```

38         containerId = p.FirstId;
39         itemId = text[2];
40         break;
41     case 5:
42         containerId = text[4];
43         itemId = text[2];
44         break;
45     }
46
47     IHaveInventory? container = FetchContainer(p, containerId);
48     if (container == null)
49         return $"I can't find the {containerId}";
50
51     return LookAtIn(itemId, container);
52 }
53
54 public IHaveInventory? FetchContainer(Player p, string containerId)
55 {
56     return p.Locate(containerId) as IHaveInventory;
57 }
58
59 public string LookAtIn(string thingId, IHaveInventory container)
60 {
61     GameObject? thing = container.Locate(thingId);
62     if (thing == null)
63         return $"I can't find the {thingId}";
64
65     return thing.FullDescription;
66 }
67 }
68 }

```

Week_7\7.2\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, IHaveInventory
9      {
10         private Inventory _inventory;
11         private Location? _location;
12
13         public Player(string name, string desc) : base(new string[] { "me", "inventory" }, name,
14 desc)
15         {
16             _inventory = new Inventory();
17         }
18     }
19 }

```



```

17
18     public GameObject? Locate(string id)
19     {
20         if (AreYou(id))
21             return this;
22
23         GameObject? obj = _inventory.Fetch(id);
24         if (obj != null)
25             return obj;
26
27         if (_location != null)
28             return _location.Locate(id);
29
30         return null;
31     }
32
33     public override string FullDescription
34     {
35         get
36         {
37             return $"You are {Name}, {base.FullDescription}\n" +
38                 $"You are carrying:\n{_inventory.ItemList}";
39         }
40     }
41
42     public Inventory Inventory => _inventory;
43     public Location? Location { get => _location; set => _location = value; }
44 }
45 }

```

Week_7\7.2\SwinAdventure\Program.cs

```

1  namespace SwinAdventure
2  {
3      class Program
4      {
5          static void Main()
6          {
7              string playerName, playerDesc;
8              while (true)
9              {
10                 Console.Write("Enter player name: ");
11                 playerName = Console.ReadLine() ?? string.Empty;
12                 Console.Write("Enter player description: ");
13                 playerDesc = Console.ReadLine() ?? string.Empty;
14                 if (string.IsNullOrEmpty(playerName) || string.IsNullOrEmpty(playerDesc))
15                 {
16                     Console.WriteLine("Player name and description cannot be empty.");
17                 }
18                 else
19                 {

```

```

20         break;
21     }
22 }
23 Player player = new Player(playerName, playerDesc);
24
25 // Create items and put them in the player's inventory
26 Item item1 = new Item(new string[] { "shovel" }, "a shovel", "a wooden shovel");
27 Item item2 = new Item(new string[] { "sword" }, "a sword", "a steel sword");
28 player.Inventory.Put(item1);
29 player.Inventory.Put(item2);
30
31 // Create a bag and put it in the player's inventory
32 Bag bag = new Bag(new string[] { "bag" }, "a bag", "a leather bag");
33 player.Inventory.Put(bag);
34
35 // Create items and put them in the bag's inventory
36 Item item3 = new Item(new string[] { "coin" }, "a coin", "a shiny coin");
37 bag.Inventory.Put(item3);
38
39 // Create location and put some items in its inventory
40 Location location = new Location("forest", "A dark forest with tall trees");
41 Item item4 = new Item(new string[] { "rock" }, "a rock", "a big rock");
42 Item item5 = new Item(new string[] { "flower" }, "a flower", "a red flower");
43 location.Inventory.Put(item4);
44 location.Inventory.Put(item5);
45
46 // Set player's location
47 player.Location = location;
48
49 LookCommand look = new LookCommand();
50
51 while (true)
52 {
53     Console.WriteLine(player.FullDescription);
54     Console.Write("> ");
55     string command = Console.ReadLine() ?? string.Empty;
56
57     if (string.IsNullOrEmpty(command))
58         continue;
59     if (command == "quit")
60         break;
61
62     string response = look.Execute(player, command.Split(" "));
63     Console.WriteLine(response);
64     Console.WriteLine();
65 }
66 }
67 }
68 }

```

```
1 using SwinAdventure;
2 using NUnit.Framework;
3
4 namespace TestLocation
5 {
6     public class TestLocation
7     {
8         private Location _location;
9         private Item _item1;
10        private Item _item2;
11        private Player _player;
12
13        [SetUp]
14        public void Setup()
15        {
16            _player = new Player("Minh An", "104844794");
17            _location = new Location("forest", "A dark forest with tall trees");
18            _player.Location = _location;
19
20            _item1 = new Item(new string[] { "rock" }, "a rock", "a big rock");
21            _item2 = new Item(new string[] { "flower" }, "a flower", "a red flower");
22            _location.Inventory.Put(_item1);
23            _location.Inventory.Put(_item2);
24        }
25
26        [Test]
27        public void LocationLocateItself()
28        {
29            Assert.That(_location.Locate("location"), Is.EqualTo(_location));
30        }
31
32        [Test]
33        public void LocationLocateItem()
34        {
35            Assert.That(_location.Locate("rock"), Is.EqualTo(_item1));
36            Assert.That(_location.Locate("flower"), Is.EqualTo(_item2));
37        }
38
39        [Test]
40        public void LocationLocateNothing()
41        {
42            Assert.That(_location.Locate("nothing"), Is.Null);
43        }
44
45        [Test]
46        public void LocationFullDescription()
47        {
48            Assert.That(_location.FullDescription, Is.EqualTo("You are in the forest.\nA dark
forest with tall trees\n" +
```

```

49 | see:\n" +
50 |
51 |
52 |     }
53 |
54 |     [Test]
55 |     public void PlayerLocateItemInLocation()
56 |     {
57 |         Assert.That(_player.Locate("rock"), Is.EqualTo(_item1));
58 |         Assert.That(_player.Locate("flower"), Is.EqualTo(_item2));
59 |     }
60 | }
61 | }

```

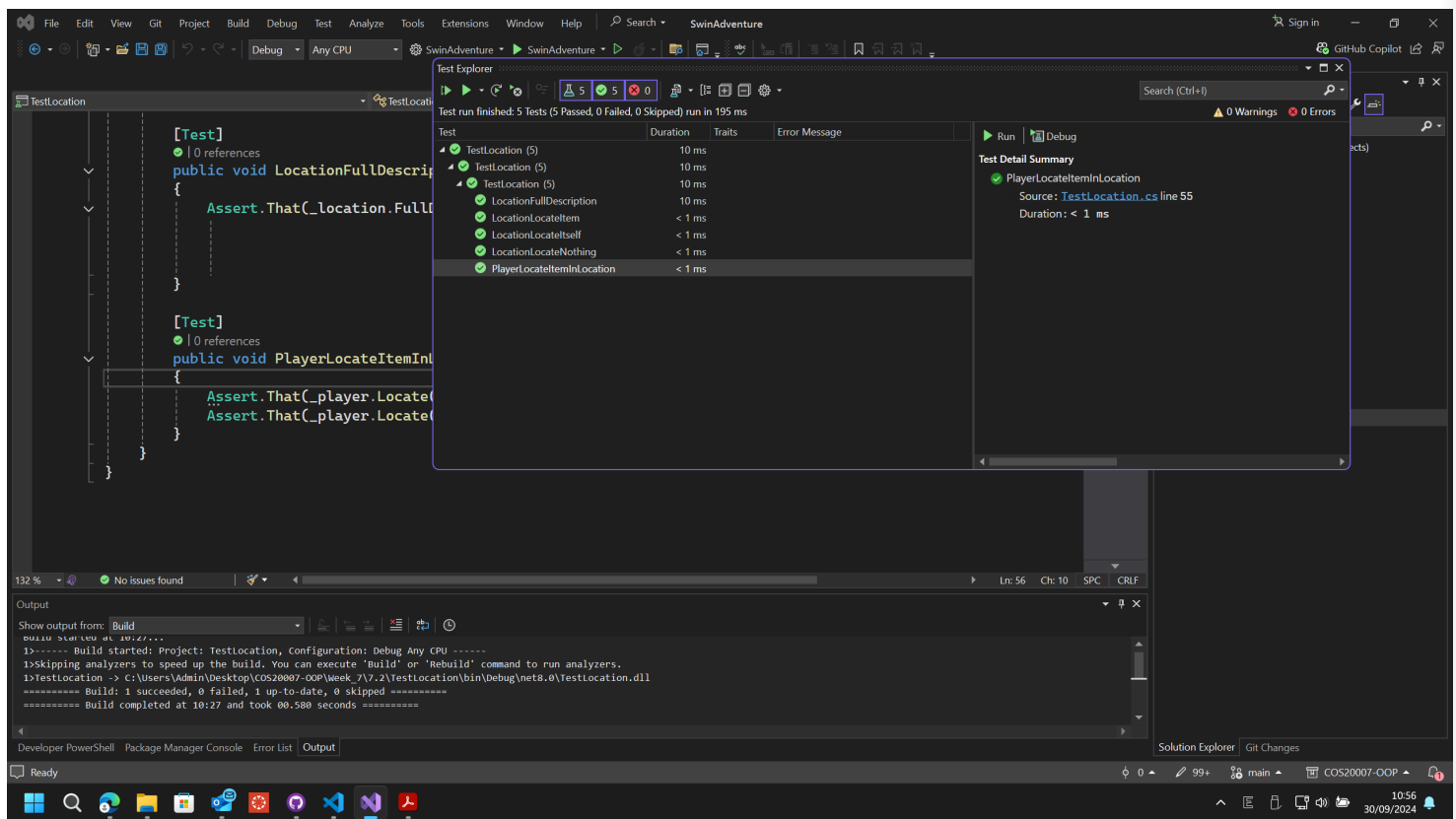
"In this location, you can

```

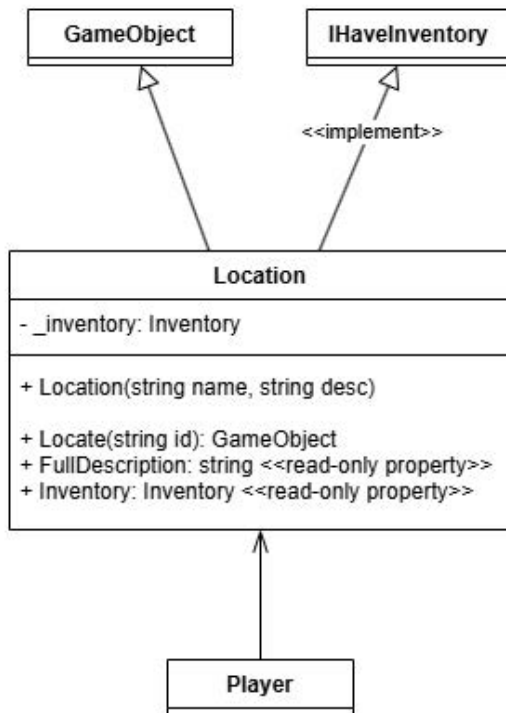
"\ta rock (rock)\n" +
"\ta flower (flower)"));

```

Screenshot of test



UML diagram



Sequence diagram

