## Selected files

#### 12 printable files

31 32

33

```
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/Program.cs
Week_7/7.2/SwinAdventure/Player.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;
   using System.Security.Cryptography;
   using System.Threading.Tasks;
 6
 7
   namespace SwinAdventure
 8
   {
 9
        public class Bag : Item, IHaveInventory
10
11
            private Inventory _inventory;
            public Bag(string[] idents, string name, string desc) : base(idents, name,
12
    desc)
            {
13
14
                _inventory = new Inventory();
            }
15
16
            public GameObject? Locate(string id)
17
18
            {
                if (AreYou(id))
19
                    return this:
20
21
22
                if (_inventory.HasItem(id))
23
                    return _inventory.Fetch(id);
24
25
                return null;
26
            }
27
28
            public Inventory Inventory
29
30
                get
```

return \_inventory;

}

```
34
35
            public override string FullDescription
36
37
                get
38
                 {
39
                     return $"In the {Name} you can see:\n{_inventory.ItemList}";
                 }
40
41
            }
42
        }
43 }
Week_7/7.2/SwinAdventure/Command.cs
 1
    using System;
   using System.Collections.Generic;
 2
    using System.Linq;
 3
    using System.Threading.Tasks;
 4
 5
 6
   namespace SwinAdventure
 7
    {
        public abstract class Command : IdentifiableObject
 8
 9
            public Command(string[] ids) : base(ids)
10
11
            {
            }
12
13
            public abstract string Execute(Player p, string[] text);
14
        }
15
16 }
Week_7/7.2/SwinAdventure/GameObject.cs
 1 using System;
   using System.Collections.Generic;
 2
    using System.Ling;
 3
    using System.Threading.Tasks;
 4
 5
    namespace SwinAdventure
 6
 7
    {
        public abstract class GameObject : IdentifiableObject
 8
 9
        {
            private string _description, _name;
10
11
            public GameObject(string[] idents, string name, string desc) : base(idents)
12
            {
13
14
                _name = name;
15
                _description = desc;
            }
16
17
18
            public string Name
19
            {
20
                get
```

```
21
                 {
22
                      return _name;
23
                 }
             }
24
25
26
             public string ShortDescription
27
             {
28
                 get
29
                 {
30
                      return $"{Name} ({FirstId})";
                 }
31
             }
32
33
34
             public virtual string FullDescription
35
36
                 get
                 {
37
                      return _description;
38
39
                 }
40
             }
41
        }
42 }
```

### Week\_7/7.2/SwinAdventure/IdentifiableObject.cs

```
1 using System;
   using System.Collections.Generic;
 2
   using System.Linq;
   using System.Threading.Tasks;
 4
 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
            public bool AreYou(string id)
21
22
            {
                return _identifiers.Contains(id.ToLower());
23
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
             public void PrivilegeEscalation(string pin)
44
45
46
                 if (pin != "4794")
47
                     return;
48
49
                 if (_identifiers.Count == 0)
50
51
                     AddIdentifier("12");
                 }
52
53
                 else
54
                 {
                     _identifiers[0] = "12";
55
                 }
56
57
            }
58
        }
59 }
```

### Week\_7/7.2/SwinAdventure/IHaveInventory.cs

```
1 using System;
 2 using System.Collections.Generic;
   using System.Linq;
   using System.Threading.Tasks;
 4
 5
6
   namespace SwinAdventure
7
   {
8
        public interface IHaveInventory
9
        {
            public GameObject? Locate(string id);
10
            public string Name { get; }
11
12
        }
13
   }
```

#### Week\_7/7.2/SwinAdventure/Inventory.cs

```
1 using System;
```

```
2 using System.Collections.Generic;
   using System.Ling;
 3
 4
   using System.Threading.Tasks;
 5
 6
   namespace SwinAdventure
 7
   {
 8
        public class Inventory : GameObject
 9
            private List<Item> _items;
10
11
            public Inventory() : base(new string[] { "inventory" }, "inventory", "The
12
    player's inventory")
13
            {
14
                _items = new List<Item>();
15
            }
16
17
            public string ItemList
18
19
                get
20
                {
                     List<string> itemsDesc = new List<string>();
21
22
                     foreach (Item item in _items)
23
                         itemsDesc.Add("\t" + item.ShortDescription);
24
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
            public void Put(Item itm)
42
43
44
                _items.Add(itm);
45
            }
46
47
            public Item? Take(string id)
48
            {
49
                foreach (Item item in _items)
50
                {
                     if (item.AreYou(id))
51
```

```
52
                     {
53
                         _items.Remove(item);
54
                         return item;
                     }
55
56
                }
57
                return null;
58
            }
59
            public Item? Fetch(string id)
60
61
            {
                foreach (Item item in _items)
62
63
                     if (item.AreYou(id))
64
65
66
                         return item;
67
                }
68
69
                return null;
70
            }
71
        }
72 }
Week_7/7.2/SwinAdventure/Item.cs
 1
   using System;
    using System.Collections.Generic;
   using System.Ling;
 3
    using System.Threading.Tasks;
 5
   namespace SwinAdventure
 6
 7
        public class Item : GameObject
 8
 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;
    using System.Linq;
    using System.Threading.Tasks;
 5
    namespace SwinAdventure
 6
 7
 8
        public class Location : GameObject, IHaveInventory
        {
 9
            private Inventory _inventory;
10
11
```

```
12
            public Location(string name, string desc) : base(new string[] { "location" },
    name, desc)
            {
13
14
                 _inventory = new Inventory();
15
            }
16
17
            public GameObject? Locate(string id)
18
19
                 if (AreYou(id))
                 {
20
21
                     return this;
22
                 }
23
24
                 return _inventory.Fetch(id);
25
            }
26
27
            public override string FullDescription
28
29
                 get
30
                 {
                     return $"You are in the {Name}.\n{base.FullDescription}\nIn this
31
    location, you can see:\n{_inventory.ItemList}";
32
            }
33
34
35
            public Inventory Inventory
36
37
                 get
38
                 {
39
                     return _inventory;
40
                 }
41
            }
42
        }
43 }
Week_7/7.2/SwinAdventure/LookCommand.cs
 1 using System;
```

```
2
   using System.Collections.Generic;
 3
   using System.Linq;
   using System.Threading.Tasks;
4
 5
   namespace SwinAdventure
6
7
8
        public class LookCommand : Command
9
        {
            public LookCommand() : base(new string[] { "look" })
10
            {
11
            }
12
13
14
            public override string Execute(Player p, string[] text)
15
            {
```

```
16
                // If text length is not 1,3,5
17
                if (text.Length != 1 && text.Length != 3 && text.Length != 5)
18
                    return "I don't know how to look like that";
19
20
                if (text[0] != "look")
21
                    return "Error in look input";
22
23
                if (text.Length != 1 && text[1] != "at")
24
                    return "What do you want to look at?";
25
                if (text.Length == 5 && text[3] != "in")
26
                    return "What do you want to look in?";
27
28
29
                string containerId = "";
30
                string itemId = "";
31
                switch (text.Length)
32
33
                    case 1:
34
                        containerId = "location";
35
                        itemId = "location";
36
                        break;
37
                    case 3:
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
                IHaveInventory? container = FetchContainer(p, containerId);
47
                if (container == null)
48
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
            public string LookAtIn(string thingId, IHaveInventory container)
59
            {
60
                GameObject? thing = container.Locate(thingId);
61
62
                if (thing == null)
                    return $"I can't find the {thingId}";
63
64
65
                return thing.FullDescription;
            }
66
```

```
67 }
68 }
```

#### Week\_7/7.2/SwinAdventure/Program.cs

```
1
   namespace SwinAdventure
 2
    {
 3
        class Program
        {
 4
            static void Main()
 5
 6
 7
                string playerName, playerDesc;
                while (true)
 8
 9
                    Console.Write("Enter player name: ");
10
11
                    playerName = Console.ReadLine();
12
                    if (playerName == null)
13
                    {
14
                         playerName = string.Empty;
15
                    }
16
                    Console.Write("Enter player description: ");
17
                    playerDesc = Console.ReadLine();
18
                    if (playerDesc == null)
19
20
21
                         playerDesc = string.Empty;
22
23
                    if (string.IsNullOrEmpty(playerName) ||
    string.IsNullOrEmpty(playerDesc))
                    {
24
25
                         Console.WriteLine("Player name and description cannot be
    empty.");
                    }
26
27
                    else
28
                    {
29
                         break:
                    }
30
                }
31
32
                Player player = new Player(playerName, playerDesc);
33
34
                // Create items and put them in the player's inventory
                Item item1 = new Item(new string[] { "shovel" }, "a shovel", "a wooden
35
    shovel");
                Item item2 = new Item(new string[] { "sword" }, "a sword", "a steel
36
    sword");
37
                player.Inventory.Put(item1);
38
                player.Inventory.Put(item2);
39
40
                // Create a bag and put it in the player's inventory
                Bag bag = new Bag(new string[] { "bag" }, "a bag", "a leather bag");
41
42
                player.Inventory.Put(bag);
43
```

```
44
                // Create items and put them in the bag's inventory
45
                Item item3 = new Item(new string[] { "coin" }, "a coin", "a shiny coin");
46
                bag.Inventory.Put(item3);
47
48
                // Create location and put some items in its inventory
49
                Location location = new Location("forest", "A dark forest with tall
    trees");
                Item item4 = new Item(new string[] { "rock" }, "a rock", "a big rock");
50
                Item item5 = new Item(new string[] { "flower" }, "a flower", "a red
51
    flower");
                location.Inventory.Put(item4);
52
                location.Inventory.Put(item5);
53
54
55
                // Set player's location
                player.Location = location;
56
57
58
                LookCommand look = new LookCommand();
59
                while (true)
60
61
                    Console.WriteLine(player.FullDescription);
62
                    Console.Write("> ");
63
64
                    string command = Console.ReadLine();
65
                    if (string.IsNullOrEmpty(command))
66
67
                         continue;
                    if (command == "quit")
68
69
                         break;
70
71
                    string response = look.Execute(player, command.Split(" "));
72
                    Console.WriteLine(response);
73
                    Console.WriteLine();
74
                }
75
            }
76
        }
77 \ }
Week_7/7.2/SwinAdventure/Player.cs
```

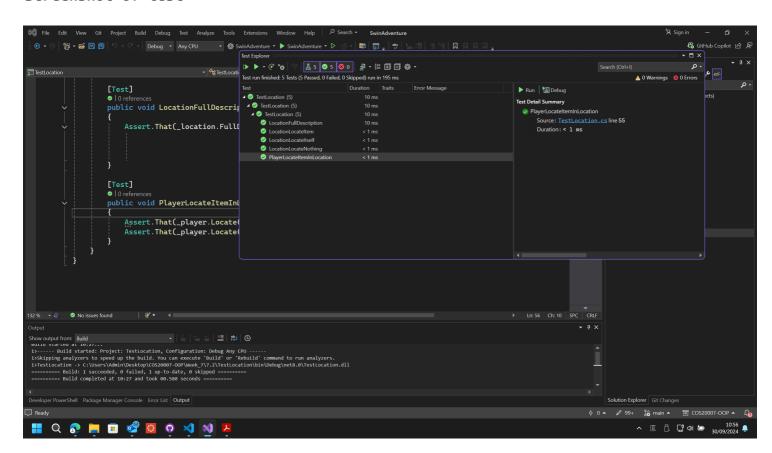
```
1 using System;
2 using System.Collections.Generic;
 3 using System.Ling;
  using System.Threading.Tasks;
5
6
  namespace SwinAdventure
7
   {
       public class Player : GameObject, IHaveInventory
8
9
10
            private Inventory _inventory;
            private Location? _location;
11
12
            public Player(string name, string desc) : base(new string[] { "me",
13
```

```
"inventory" }, name, desc)
14
            {
15
                 _inventory = new Inventory();
            }
16
17
            public GameObject? Locate(string id)
18
19
                 if (AreYou(id))
20
21
                     return this;
22
23
                 GameObject? obj = _inventory.Fetch(id);
                 if (obj != null)
24
                     return obj;
25
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
                 {
                     return $"You are {Name}, {base.FullDescription}\n" +
37
                             $"You are carrying:\n{_inventory.ItemList}";
38
39
                 }
            }
40
41
42
            public Inventory Inventory
43
44
                 get
45
                 {
                     return _inventory;
46
                 }
47
            }
48
49
            public Location? Location
50
51
                 get
52
                 {
                     return _location;
53
54
                 }
55
                 set
56
                 {
57
                     _location = value;
58
                 }
59
            }
60
        }
```

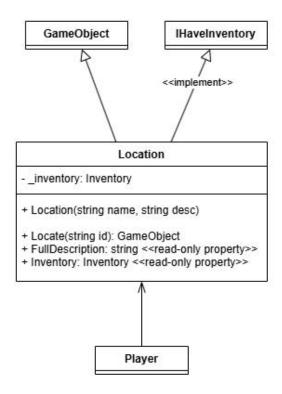
```
1 using SwinAdventure;
 2
   using NUnit.Framework;
 3
 4
   namespace TestLocation
 5
   {
        public class TestLocation
 6
 7
        {
 8
            private Location _location;
 9
            private Item _item1;
            private Item _item2;
10
            private Player _player;
11
12
13
            [SetUp]
            public void Setup()
14
15
                _player = new Player("Minh An", "104844794");
16
                _location = new Location("forest", "A dark forest with tall trees");
17
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
            [Test]
26
27
            public void LocationLocateItself()
28
                Assert.That(_location.Locate("location"), Is.EqualTo(_location));
29
30
            }
31
            [Test]
32
            public void LocationLocateItem()
33
34
                Assert.That(_location.Locate("rock"), Is.EqualTo(_item1));
35
                Assert.That(_location.Locate("flower"), Is.EqualTo(_item2));
36
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
                Assert.That(_location.FullDescription, Is.EqualTo("You are in the
48
    forest.\nA dark forest with tall trees\n" +
49
                                                                      "In this location, you
    can see:\n" +
```

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

#### Screenshot of test



## **UML** diagram



# Sequence diagram

