

Selected files

10 printable files

Week_7\7.1\SwinAdventure\Program.cs
Week_7\7.1\SwinAdventure\Bag.cs
Week_7\7.1\SwinAdventure\Command.cs
Week_7\7.1\SwinAdventure\GameObject.cs
Week_7\7.1\SwinAdventure\IdentifiableObject.cs
Week_7\7.1\SwinAdventure\IHaveInventory.cs
Week_7\7.1\SwinAdventure\Inventory.cs
Week_7\7.1\SwinAdventure\Item.cs
Week_7\7.1\SwinAdventure\LookCommand.cs
Week_7\7.1\SwinAdventure\Player.cs

Week_7\7.1\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             Item item1 = new Item(new string[] { "shovel" }, "a shovel", "a wooden shovel");
26             Item item2 = new Item(new string[] { "sword" }, "a sword", "a steel sword");
27             player.Inventory.Put(item1);
28             player.Inventory.Put(item2);
29
30             Bag bag = new Bag(new string[] { "bag" }, "a bag", "a leather bag");
31             player.Inventory.Put(bag);
32
33             Item item3 = new Item(new string[] { "coin" }, "a coin", "a shiny coin");
34             bag.Inventory.Put(item3);
35             LookCommand look = new LookCommand();
```

```

36
37     while (true)
38     {
39         Console.WriteLine(player.FullDescription);
40         Console.Write("> ");
41         string command = Console.ReadLine() ?? string.Empty;
42
43         if (string.IsNullOrEmpty(command))
44             continue;
45         if (command == "quit")
46             break;
47
48         string response = look.Execute(player, command.Split(" "));
49         Console.WriteLine(response);
50         Console.WriteLine();
51     }
52 }
53 }
54 }

```

Week_7\7.1\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 }

```

Week_7\7.1\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\7.1\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\7.1\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\7.1\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\7.1\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 }
```

Week_7\7.1\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, desc)
11         {
12         }
13     }
14 }
```

Week_7\7.1\SwinAdventure\LookCommand.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 LookCommand : Command
9     {
10         public LookCommand() : base(new string[] { "look" })
11         {
12         }
13
14         public override string Execute(Player p, string[] text)
15         {
16             if (text.Length != 3 && text.Length != 5)
17                 return "I don't know how to look like that";
18
19             if (text[0] != "look")
20                 return "Error in look input";
21
22             if (text[1] != "at")
23                 return "What do you want to look at?";
24
25             if (text.Length == 5 && text[3] != "in")
26                 return "What do you want to look in?";
27
28             string containerId = "";
29             if (text.Length == 3)
```

```

30         containerId = p.FirstId;
31     else if (text.Length == 5)
32         containerId = text[4];
33
34     IHaveInventory? container = FetchContainer(p, containerId);
35     if (container == null)
36         return $"I can't find the {containerId}";
37
38     return LookAtIn(text[2], container);
39 }
40
41 public IHaveInventory? FetchContainer(Player p, string containerId)
42 {
43     return p.Locate(containerId) as IHaveInventory;
44 }
45
46 public string LookAtIn(string thingId, IHaveInventory container)
47 {
48     GameObject? thing = container.Locate(thingId);
49     if (thing == null)
50         return $"I can't find the {thingId}";
51
52     return thing.FullDescription;
53 }
54 }
55 }

```

Week_7\7.1\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
12         public Player(string name, string desc) : base(new string[] { "me", "inventory" }, name,
13             desc)
14         {
15             _inventory = new Inventory();
16         }
17
18         public GameObject? Locate(string id)
19         {
20             if (AreYou(id))
21                 return this;

```



```

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 }

```

Screenshot of running program:

```

Week_7 > 7.1 > SwinAdventure > C# Program.cs > ...
3     class Program
5     static void Main()

29
30
31
32
33
34
35
36
37
38
39

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL TEST RESULTS COMMENTS

```

PS C:\Users\Admin\Desktop\COS20007-OOP\Week_7\7.1\SwinAdventure> dotnet run
Enter player name: Minh An Nguyen
Enter player description: 104844794
You are Minh An Nguyen, 104844794
You are carrying:
    a shovel (shovel)
    a sword (sword)
    a bag (bag)
> look at shovel
a wooden shovel

You are Minh An Nguyen, 104844794
You are carrying:
    a shovel (shovel)
    a sword (sword)
    a bag (bag)
> look at sword
a steel sword

You are Minh An Nguyen, 104844794
You are carrying:
    a shovel (shovel)
    a sword (sword)
    a bag (bag)
> look at bag
In the a bag you can see:
    a coin (coin)

You are Minh An Nguyen, 104844794
You are carrying:
    a shovel (shovel)
    a sword (sword)
    a bag (bag)
> look at coin in bag
a shiny coin

```