

COS20007 - Object Oriented Programming

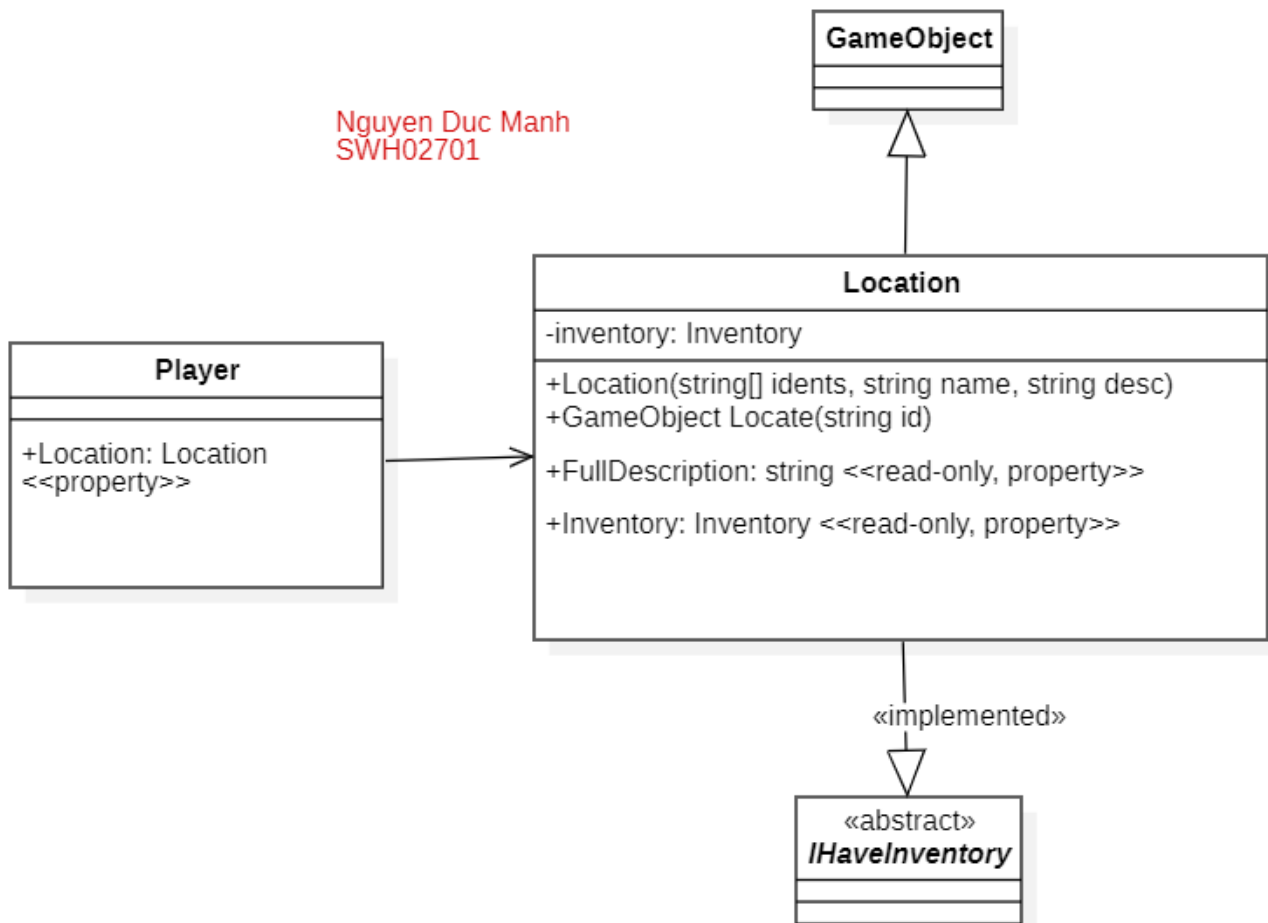
7.2C - Case Study — Advanced Iteration 6: Locations

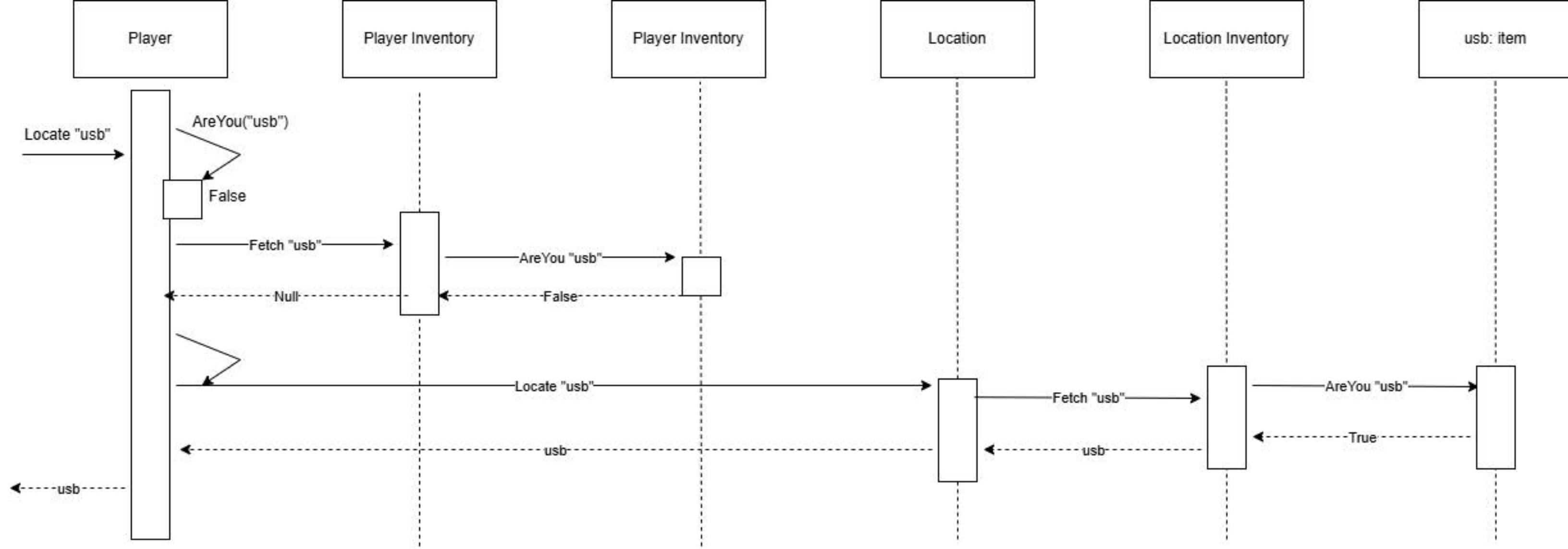
Student name: Nguyen Duc Manh
ID: 105547489



UML, Sequence diagram and NUnit Test

Nguyen Duc Manh
SWH02701





```
1 namespace SwinAdventure;
2
3 public class LocationTest
4 {
5     Location _location;
6     Player _player;
7
8     [SetUp]
9     public void Setup()
10    {
11        _location = new(["duytan", "location"], "80 duy tan", "Innovation
12                        Space");
13        _player = new("Manh", "A student at Swinburne");
14    }
15
16    [Test]
17    public void LocationLocateItself()
18    {
19        Assert.IsTrue(_location.AreYou("duytan"));
20    }
21
22    [Test]
23    public void LocationLocateItem()
24    {
25        Item item = new(["usb", "an usb", "this is an usb"]);
26        _location.Inventory.Put(item);
27        Assert.IsTrue(_location.Locate("usb").AreYou("usb"));
28    }
29
30    [Test]
31    public void PlayerInLocation()
32    {
33        _player.Location = _location;
34        string expect = _location.FullDescription;
35        Assert.That(_player.Location.FullDescription, Is.EqualTo(expect));
36    }
37
38    [Test]
39    public void PlayerLocateLocation()
40    {
41        _player.Location = _location;
42        Assert.IsTrue(_player.Locate("duytan").AreYou("duytan"));
43    }
44
45    [Test]
46    public void PlayerLocateItem()
47    {
48        Item item = new(["usb", "an usb", "magical usb can stores 1TB"]);
49        _location.Inventory.Put(item);
```

```
49         _player.Location = _location;
50         Assert.IsTrue(_player.Locate("usb").AreYou("usb"));
51     }
52 }
53
```

Test	Duration	Traits	Error Message
ObjTest (33)	199 ms		
SwinAdventure (33)	199 ms		
BagsTest (5)	192 ms		
InventoryTest (5)	2 ms		
ItemTest (4)	< 1 ms		
LocationTest (5)	1 ms		
LocationLocateItem	1 ms		
LocationLocateItself	< 1 ms		
PlayerInLocation	< 1 ms		
PlayerLocateItem	< 1 ms		
PlayerLocateLocation	< 1 ms		
LookCommandTest (9)	2 ms		
PlayerTest (5)	2 ms		

Run | Debug

Group Summary

ObjTest

Tests in group: 33

Total Duration: 199 ms

Outcomes

33 Passed

Source Code


```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SwinAdventure
8 {
9     public class Location : GameObject, IHaveInventory
10     {
11         Inventory _inventory;
12
13         public Location(string[] idents, string name, string desc) : base
14             (idents, name, desc)
15         {
16             _inventory = new Inventory();
17         }
18
19         public GameObject Locate(string id)
20         {
21             if (AreYou(id))
22             {
23                 return this;
24             }
25             return _inventory.Fetch(id);
26         }
27
28         public override string FullDescription
29         {
30             get
31             {
32                 return $"You are in: {Name}, {base.FullDescription}. Here
33                     you can see: {_inventory.ItemList()}";
34             }
35         }
36
37         public Inventory Inventory
38         {
39             get { return _inventory; }
40         }
41     }
```

```
1 namespace SwinAdventure
2 {
3     public class Program
4     {
5         public static void Main(string[] args)
6         {
7             Console.WriteLine("Enter your player's name: ");
8             string name = Console.ReadLine();
9
10            Console.WriteLine("Enter player's description: ");
11            string des = Console.ReadLine();
12
13            // Create a player and some items
14            Player player = new(name, des);
15            Item itm1 = new(["hdmi"], "HDMI cord", "can connect to large screen");
16            Item itm2 = new(["usb"], "an USB", "can store up to 1TB of data");
17            Bags bag = new(["bag"], "a bag", "this bag is made by leather");
18            Item itm3 = new(["mouse"], "a mouse", "gaming mouse with 0 latency");
19            Command lookCommand = new LookCommand();
20            Location duytan = new(["duytan"], "duytan", "Innovation Center of Swinburne");
21
22            player.Inventory.Put(itm1);
23            player.Inventory.Put(itm2);
24            player.Inventory.Put(bag);
25            bag.Inventory.Put(itm3);
26            duytan.Inventory.Put(itm1);
27            player.Location = duytan;
28
29            //loop command
30            while (true)
31            {
32                Console.WriteLine("Enter what do you want to find:");
33                string userInput = Console.ReadLine();
34                if (userInput.ToLower() != "exit")
35                {
36                    if (userInput == "look")
37                    {
38                        string[] userCommand = [userInput];
39                        string result = lookCommand.Execute(player, userCommand);
40                        Console.WriteLine(" ");
41                        Console.WriteLine(result);
42                    }
43                    else
```

```
44         {
45             string[] userCommand = userInput.Split(' ');
46             string result = lookCommand.Execute(player, userCommand);
47             Console.WriteLine(" ");
48             Console.WriteLine($"{result}\n");
49         }
50     }
51     else break;
52 }
53 Console.WriteLine("Iteration 5 finished !");
54 }
55 }
56 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7 using static System.Runtime.InteropServices.JavaScript.JSType;
8 using System.Xml.Linq;
9
10 namespace SwinAdventure
11 {
12     public class LookCommand : Command
13     {
14         IHaveInventory container;
15         GameObject item;
16         Player p;
17         Location location;
18
19         public LookCommand() : base(["look"]) { }
20
21         public override string Execute(Player p, string[] text)
22         {
23             if (text.Length == 1 && text[0].ToLower() == "look")
24             {
25                 return p.Location.FullDescription;
26             }
27             if (text.Length == 3 || text.Length == 5)
28             {
29                 if (text[0] != "look")
30                     return "Error in look input";
31                 if (text[1] != "at")
32                     return "What do you want to look at?";
33                 if (text.Length == 5 && text[3] != "in")
34                     return "What do you want to look in?";
35                 if (text.Length == 3)
36                 {
37                     container = p;
38                 }
39                 else
40                 {
41                     container = FetchContainer(p, text[4]);
42                     if (container == null)
43                         return $"I cannot find the {text[4]}";
44                 }
45
46                 return LookAtIn(text[2], container);
47             }
48             else
49                 return "I don't know how to look like that";
```

```
50     }
51
52     private IHaveInventory? FetchContainer(Player p, string containerId)
53     {
54         return p.Locate(containerId) as IHaveInventory;
55     }
56
57     private string LookAtIn(string thingId, IHaveInventory container)
58     {
59         if (container.Locate(thingId) != null)
60         {
61             return container.Locate(thingId).FullDescription;
62         }
63         else
64             return $"I cannot find the {thingId}";
65     }
66 }
67 }
68
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SwinAdventure
8 {
9     public class Player : GameObject, IHaveInventory
10     {
11         Inventory _inventory;
12
13         public Player (string name, string desc) : base(new string[] { "me", "inventory"}, name, 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             var itm = _inventory.Fetch(id);
25             if (itm != null)
26             {
27                 return itm;
28             }
29             if (Location != null)
30             {
31                 return Location.Locate(id);
32             }
33             return null;
34         }
35
36         public override string FullDescription
37         {
38             get
39             {
40                 return $"{Name}, {base.ShortDescription}. You are carrying:
41                     {_inventory.ItemList()}";
42             }
43         }
44
45         public Inventory Inventory { get { return _inventory; } }
46
47         public Location Location { get; set; }
```

48 }

49

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SwinAdventure
8 {
9     public abstract class Command : IdenObj
10    {
11        public Command(string[] ids) : base(ids)
12        {
13            //
14        }
15
16        public abstract string Execute(Player p, string[] text);
17    }
18 }
19
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SwinAdventure
8 {
9     public interface IHaveInventory
10    {
11        public GameObject Locate(string id);
12
13        public string Name { get; }
14    }
15 }
16
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SwinAdventure
8 {
9     public class Bags : Item, IHaveInventory
10    {
11        Inventory _inventory;
12
13        public Bags(string[] idents, string name, string desc) : base
14            (idents, name, desc)
15        {
16            _inventory = new Inventory();
17        }
18
19        public GameObject Locate(string id)
20        {
21            if (AreYou(id))
22            {
23                return this;
24            }
25            else if (_inventory.HasItem(id))
26            {
27                return _inventory.Fetch(id);
28            }
29            return null;
30        }
31
32        public override string FullDescription
33        {
34            get { return $"In the {Name} you can see:\n{_inventory.ItemList
35                ()}" ; }
36        }
37
38        public Inventory Inventory
39        {
40            get { return _inventory; }
41        }
42    }
43 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SwinAdventure
8 {
9     public abstract class GameObject : IdenObj
10    {
11        string _description;
12        string _name;
13
14        public GameObject(string[] idents, string name, string desc) : base ↗
15            (idents)
16        {
17            _name = name;
18            _description = desc;
19        }
20
21        public string Name { get { return _name; } }
22
23        public string ShortDescription { get { return $"{_name} ↗
24            ({FirstId})"; } }
25
26        public virtual string FullDescription { get { return ↗
27            _description; } }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SwinAdventure
8 {
9     public class IdenObj
10    {
11        //fields
12        private List<string> _identifiers;
13        string _myStudentID = "7489";
14
15        //constructor
16        public IdenObj(string[] idents)
17        {
18            _identifiers = new List<string>();
19            if (idents != null)
20            {
21                for (int i = 0; i < idents.Length; i++)
22                {
23                    _identifiers.Add(idents[i].ToLower());
24                }
25            }
26        }
27
28        //methods
29        public bool AreYou(string id)
30        {
31            return _identifiers.Contains(id.ToLower());
32        }
33
34        public string FirstId
35        {
36            get
37            {
38                if( _identifiers.Count == 0)
39                {
40                    return "";
41                } else { return _identifiers.First(); }
42            }
43        }
44
45        public void AddIdentifier(string id)
46        {
47            _identifiers.Add(id.ToLower());
48        }
49    }
```

```
50     public void PrivilegeEscalation(string pin)
51     {
52         if(pin.Length == 4)
53         {
54             if(pin == _myStudentID) //105547489
55             {
56                 _identifiers[0] = _myStudentID;
57             }
58         }
59         else
60         {
61             return;
62         }
63     }
64 }
65 }
66 }
67 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SwinAdventure
8 {
9     public class Item : GameObject
10    {
11        public Item(string[] idents, string name, string desc) : base
12            (idents, name, desc)
13        {
14            //not yet
15        }
16    }
17 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SwinAdventure
8 {
9     public class Inventory
10    {
11        List<Item> _items;
12
13        public Inventory()
14        {
15            _items = new List<Item>();
16        }
17
18        public bool HasItem(string id)
19        {
20            foreach (Item item in _items)
21            {
22                if (item.AreYou(id))
23                {
24                    return true;
25                }
26            }
27            return false;
28        }
29
30        public void Put(Item itm)
31        {
32            _items.Add(itm);
33        }
34
35        //public void RemoveItm(Item itm)
36        //{
37        //    if (_items.Contains(itm))
38        //    {
39        //        _items.Remove(itm);
40        //    }
41        //}
42
43        public Item Take(string id)
44        {
45            foreach (Item item in _items)
46            {
47                if (item.AreYou(id))
48                {
49                    _items.Remove(item);
```

```
50         return item;
51     }
52 }
53     return null;
54 }
55
56     public Item Fetch(string id)
57     {
58         foreach (Item item in _items)
59         {
60             if (item.AreYou(id))
61             {
62                 return item;
63             }
64         }
65         return null;
66     }
67
68     public string ItemList()
69     {
70         string listitm = "";
71         foreach (Item item in _items)
72         {
73             listitm = listitm + item.ShortDescription + "\n";
74         }
75         return listitm;
76     }
77 }
78 }
79
```