COS20007 - Object Oriented Programming

7.1P - Case Study - Iteration 5 - Console Application typing prior iterations

Student name: Nguyen Duc Manh
ID: 105547489

```
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
21             player.Inventory.Put(itm1);
22             player.Inventory.Put(itm2);
23             player.Inventory.Put(bag);
24             bag.Inventory.Put(itm3);
25
26             //loop command
27             while (true)
28             {
29                 Console.WriteLine("Enter what do you want to find:");
30                 string userInput = Console.ReadLine();
31                 if (userInput.ToLower() != "exit")
32                 {
33                     string[] userCommand = userInput.Split(' ');
34                     string result = lookCommand.Execute(player,
                         userCommand);
35                     Console.WriteLine(" ");
36                     Console.WriteLine($"Description for {userCommand
                         [2]}:");
37                     Console.WriteLine($"{result}\n");
38                 }
39                 else break;
40             }
41             Console.WriteLine("Iteration 5 finished !");
42         }
43     }
```

```
44  }
```

```
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
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using static System.Runtime.InteropServices.JavaScript.JSType;
using System.Xml.Linq;

namespace SwinAdventure
{
    public class LookCommand : Command
    {
        IHaveInventory container;
        GameObject item;
        Player p;

        public LookCommand() : base(["look"]) { }

        public override string Execute(Player p, string[] text)
        {
            if (text.Length == 3 || text.Length == 5)
            {
                if (text[0] != "look")
                    return "Error in look input";
                if (text[1] != "at")
                    return "What do you want to look at?";
                if (text.Length == 5 && text[3] != "in")
                    return "What do you want to look in?";
                if (text.Length == 3)
                {
                    container = p;
                }
                else
                {
                    container = FetchContainer(p, text[4]);
                    if (container == null)
                        return $"I cannot find the {text[4]}";
                }

                return LookAtIn(text[2], container);
            }
            else
                return "I don't know how to look like that";
        }

        private IHaveInventory? FetchContainer(Player p, string
          containerId)
        {
```

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

```
 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 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             return _inventory.Fetch(id);
25         }
26
27         public override string FullDescription
28         {
29             get
30             {
31                 return $"{Name}, {base.ShortDescription}.You are carrying:
                     {_inventory.ItemList()}";
32             }
33         }
34
35         public Inventory Inventory { get { return _inventory; } }
36     }
37 }
38
```

```
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
             (idents, 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             else if (_inventory.HasItem(id))
25             {
26                 return _inventory.Fetch(id);
27             } return null;
28         }
29
30         public override string FullDescription
31         {
32             get { return $"In the {Name} you can see:\n{_inventory.ItemList
                 ()}" ; }
33         }
34
35         public Inventory Inventory
36         {
37             get { return _inventory; }
38         }
39     }
40 }
41
```

```
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
             (idents)
15         {
16             _name = name;
17             _description = desc;
18         }
19
20         public string Name { get { return _name; } }
21
22         public string ShortDescription { get { return $"{_name}
             ({FirstId})"; } }
23
24         public virtual string FullDescription { get { return
             _description; } }
25     }
26 }
27
```

```
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
              (idents, name, desc)
12          {
13              //not yet
14          }
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
```

```
Enter your player's name:
Manh
Enter player's description:
IT
Enter what do you want to find:
look at bag

Description for bag:
In the a bag you can see:
a mouse (mouse)


Enter what do you want to find:
look at mouse in bag

Description for mouse:
gaming mouse with 0 latency

Enter what do you want to find:
look at hdmi

Description for hdmi:
can connect to large screen

Enter what do you want to find:
exit
Iteration 5 finished !

C:\Users\Bill\Desktop\COS20007\Week 7\Iteration 5\SwinAdventure\bin\Debug\net8.0\SwinAdventure.exe (process 11336) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```