```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace IdentifiableObject
8  {
9      public class CommandHandler:Command
10     {
11         List<Command> _commands;
12
13         public CommandHandler() : base(new string[] { "command" })
14         {
15             _commands = new List<Command>();
16             _commands.Add(new LookCommand());
17             _commands.Add(new MoveCommand());
18         }
19
20         public override string Execute(Player p, string[] text)
21         {
22             foreach (Command c in _commands)
23             {
24                 if (c.AreYou(text[0].ToLower()))
25                 {
26                     return c.Execute(p, text);
27                 }
28             }
29             return "Your command is wrong";
30         }
31     }
32 }
33
```

```csharp
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace IdentifiableObject
8  {
9      public class Locations:GameObject, IHaveInventory
10     {
11         private Inventory _item_in_location;
12         private List<Path> _paths;
13
14         public Locations(string name, string desc):base(new string[]        ⮡
             {"location"}, name, desc)
15         {
16             _item_in_location = new Inventory();
17             _paths = new List<Path>();
18         }
19
20         public Locations(string name, string desc, List<Path> paths) : this ⮡
             (name, desc)
21         {
22             _paths = paths;
23         }
24
25         public GameObject Locate(string id)
26         {
27             if(AreYou(id))
28             {
29                 return this;
30             }
31             else
32             {
33                 foreach (Path p in _paths)
34                 {
35                     if (p.AreYou(id))
36                     {
37                         return p;
38                     }
39                 }
40                 return _item_in_location.Fetch(id);
41             }
42         }
43
44         public Inventory ItemInLocation
45         {
46             get
47             {
```

```
48                    return _item_in_location;
49                }
50            }
51
52        public override string FullDescription
53        {
54            get
55            {
56                return $"Welcome, {base.FullDescription}\nIn this location  ⮡
                    you can see:\n{ItemInLocation.ItemList} ";
57            }
58        }
59
60        public void AddPath(Path path)
61        {
62            _paths.Add(path);
63        }
64
65
66
67
68        public override string ShortDescription
69        {
70            get
71            {
72                return "You are in a " + Name;
73            }
74        }
75    }
76 }
77
```

```csharp
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace IdentifiableObject
8  {
9      public class MoveCommand:Command
10     {
11         public MoveCommand():base(new string[] {"move"})
12         {
13
14         }
15
16         public override string Execute(Player p, string[] text)
17         {
18
19             if(text.Length==2)
20             {
21                 if(text[0].ToLower()=="move")
22                 {
23                     GameObject path = p.Location.Locate(text[1]);
24                     if(path!=null)
25                     {
26                         if(path is not Path _path)
27                         {
28                             return "Can't find the " + path.Name;
29                         }
30                         else
31                         {
32                             p.Move((Path)path);
33                             return $"You go {path.FirstId}\nWent through
                   the {path.Name}\nArrived to the {p.Location.Name}";
34                         }
35                     }
36                     else
37                     {
38                         return "There's no path like that";
39                     }
40                 }
41                 else
42                 {
43                     return "Wrong format of the command";
44                 }
45             }
46             else
47             {
48                 return "I don't know how to move like that.";
```

```
49              }
50          }
51      }
52 }
53
```

```csharp
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace IdentifiableObject
8  {
9      public class Path:GameObject
10     {
11         private bool _block;
12         private Locations _current_location;
13         private Locations _destination;
14
15         public Path(string[] ids, string name, string desc, Locations      ⮡
              current_location, Locations destination):base(ids, name, desc)
16         {
17             _block = false;
18             _current_location = current_location;
19             _destination = destination;
20             AddIdentifier("path");
21             foreach (string s in name.Split(" "))
22             {
23                 AddIdentifier(s);
24             }
25         }
26
27         public Locations Destination
28         {
29             get
30             {
31                 return _destination;
32             }
33         }
34
35         public override string FullDescription
36         {
37             get
38             {
39                 return Name;
40             }
41         }
42
43         public bool Block
44         {
45             get
46             {
47                 return _block;
48             }
```

```
49              set
50              {
51                  _block = value;
52              }
53          }
54      }
55  }
56
```

```csharp
1  using IdentifiableObject;
2  using System;
3
4  class Program
5  {
6      static void Main(string[] args)
7      {
8          string player_name, player_desc;
9          Player player;
10         Bag player_bag=new Bag(new string[]{"bag"}, "Thang's bag", "bag
             number 104776473");
11         Command command = new CommandHandler();
12         Locations guild = new Locations("Guild", "This is the city's
             guild");
13
14         Locations shop = new Locations("equipment shop", "place to sell
             tools");
15         IdentifiableObject.Path guild_to_shop = new IdentifiableObject.Path
             (new string[] { "south" }, "Guild's Door", "Cross through the
             door", guild, shop);
16         IdentifiableObject.Path shop_to_guild = new IdentifiableObject.Path
             (new string[] { "north" }, "Shop's Door", "Cross through the
             door", shop, guild);
17         guild.AddPath(guild_to_shop);
18         shop.AddPath(shop_to_guild);
19
20         Console.Write("Please enter your name: ");
21         player_name = Console.ReadLine();
22         Console.Write("Please enter your description: ");
23         player_desc = Console.ReadLine();
24
25         player = new Player(player_name, player_desc);
26
27         Item shield = new Item(new string[] { "shield" }, "a shield",
             "Shield level 1");
28         Item shovel = new Item(new string[] { "shovel" }, "a shovel",
             "Shovel level 2");
29
30         player.Inventory.Put(shield);
31         player.Inventory.Put(shovel);
32
33         player.Inventory.Put(player_bag);
34
35         Item blanket = new Item(new string[] { "blanket" }, "a blanket",
             "winter blanket");
36         player_bag.Inventory.Put(blanket);
37
38         player.Location = guild;
39
```

```
40          Item sword = new Item(new string[] { "sword" }, "a sword", "Sword  ⮐
                level 10");
41          Item staff = new Item(new string[] { "staff" }, "a staff", "Staff  ⮐
                level 20");
42       guild.ItemInLocation.Put(sword);
43       guild.ItemInLocation.Put(staff);
44
45          Bag b2= new Bag(new string[]{ "b2"}, "bag2", "bag number 2");
46       player_bag.Inventory.Put(b2);
47
48
49          while (true)
50          {
51              Console.Write("Orders: ");
52              string user_input = Console.ReadLine();
53
54              Console.WriteLine(command.Execute(player, user_input.Split()));
55
56          }
57      }
58 }
```

```
E:\COS20007\week9\9.2C\Swi   X      +   ∨

Please enter your name: thang
Please enter your description: 104776473
Orders: move south
You go south
Went through the Guild's Door
Arrived to the equipment shop
Orders: look at shield
Shield level 1
Orders: look at blanket
I can't find the blanket
Orders: look at blanket in bag
winter blanket
Orders: |
```

```
                    ┌─────────────────────────┐
                    │    IdentifiableObject    │
                    ├─────────────────────────┤
                    │                          │
                    ├─────────────────────────┤
                    │                          │
                    └─────────────────────────┘
                                 △
                                 │
                    ┌─────────────────────────┐
                    │         Command          │
                    ├─────────────────────────┤
                    │                          │
                    ├─────────────────────────┤
                    │ + Command(string[] ids)  │
                    │                          │
                    │ + Execute: string <<abstract>> │
                    └─────────────────────────┘
```

- **IdentifiableObject**

- **Command**
  - + Command(string[] ids)
  - + Execute: string <>

- **LookCommand**

- **MoveCommand**

- **CommandHandler**
  - - _commands: List<command>
  - + CommandHandler()
  - + Execute(Player p, string[] text): string <<override>>

Use

Use

```
CommandHandler          LookCommand          MoveCommand          CommandHandler

      |                      |                    |                    |
      | Execute(p, "look", "at", "shield")         |                    |
      |--------------------->| Look At In("shield", p)                 |
      |                      |----┐                |                    |
      |                      |    |                |                    |
      |                      |<---┘                |                    |
      |                      |          Locate "shield"                 |
      |                      |------------------------------------------>|
      |                      |                     |                    |
      |                      |              shield                       |
      |                      |<- - - - - - - - - - - - - - - - - - - - - |
      |                      |                     |                    |
      | Execute(p, "move", "north")                |                    |
      |--------------------->|                     |                    |
      |                      |                     |                    |
      |                      |      Locate "north" |                    |
      |                      |-------------------->|                    |
      |                      |                     |                    |
      |                      |      north          |                    |
      |                      |<- - - - - - - - - - |                    |
      |                      |                     |                    |
```