

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

Case Study - Iteration 8 - Command Processor

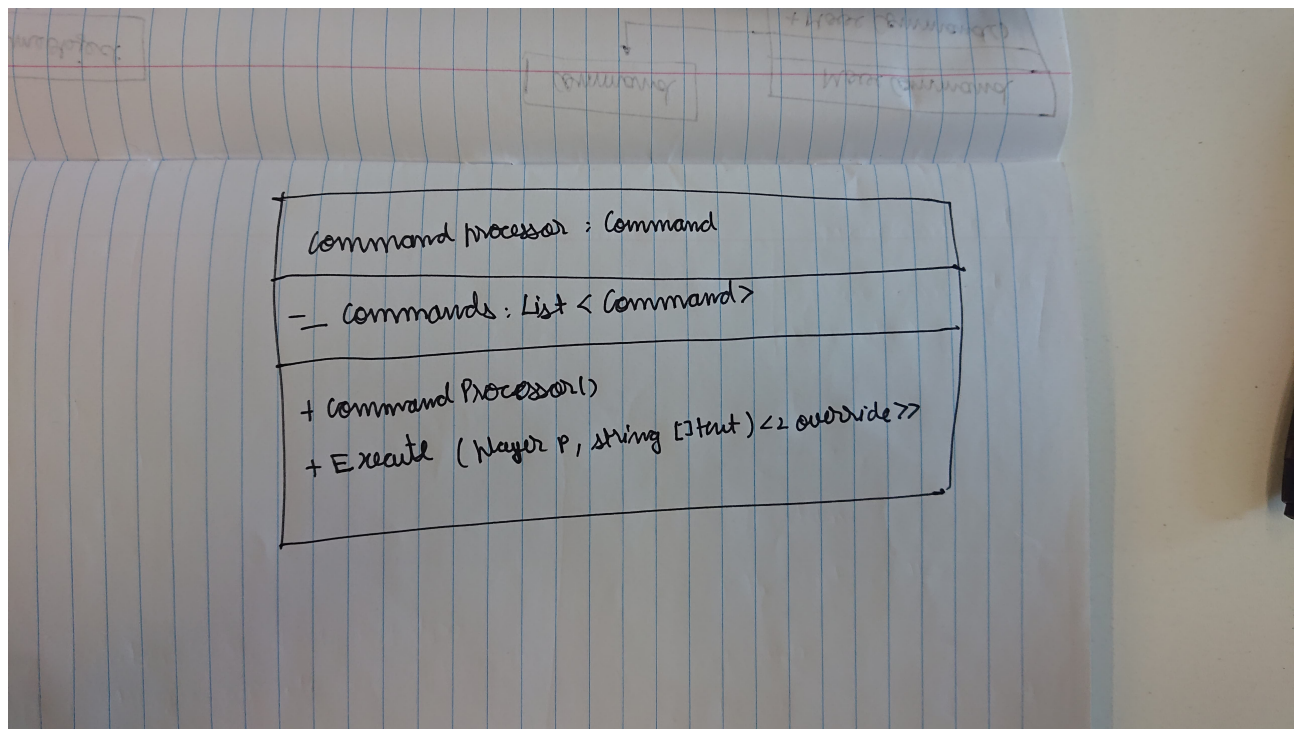
PDF generated at 12:42 on Friday 27th October, 2023

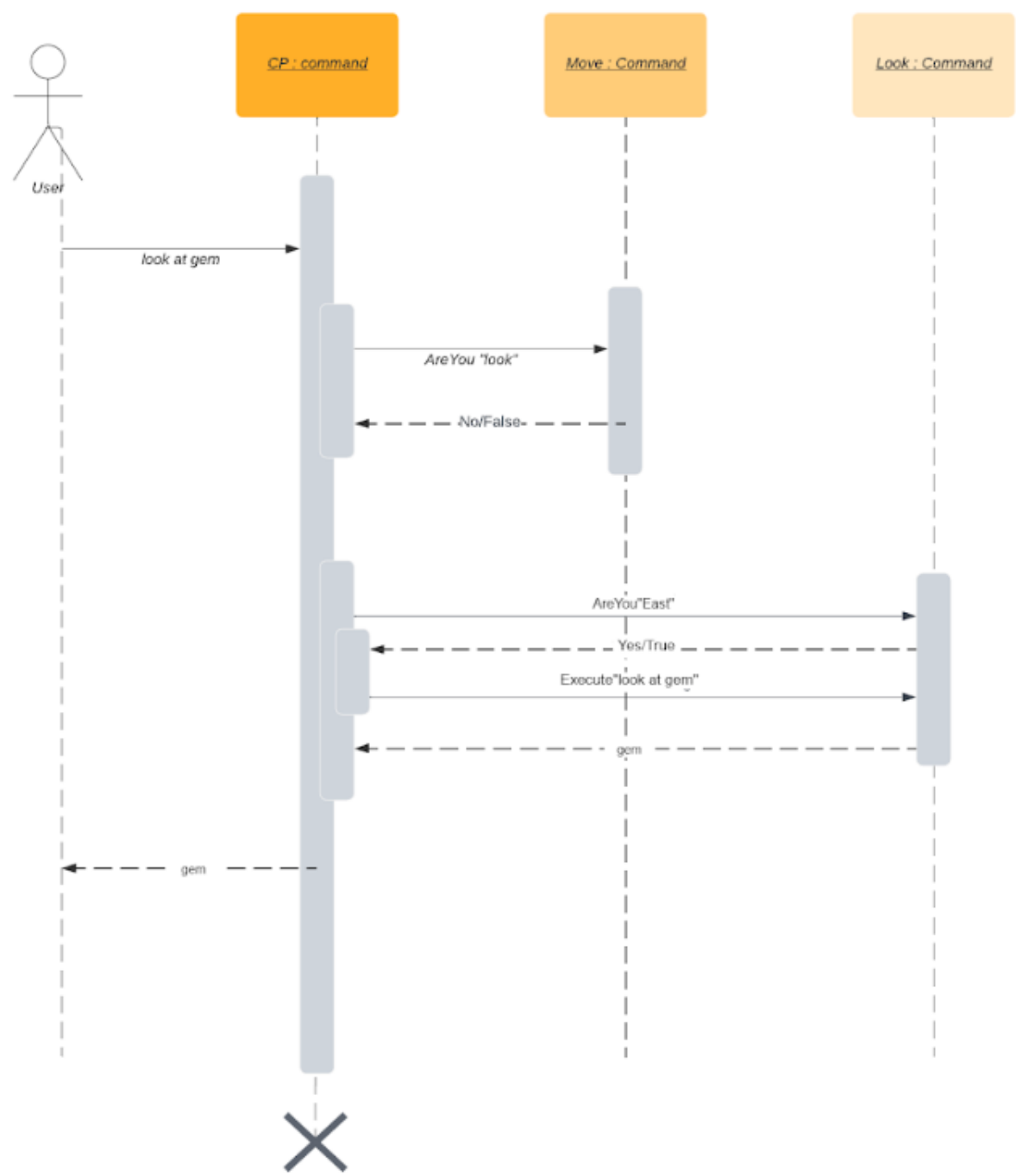
```
1
2
3 namespace Iteration2
4 {
5     class MainClass
6     {
7         public static void Main(string[] args)
8         {
9             string name;
10            string desc;
11            Player player;
12            Console.WriteLine("Enter your name:");
13            name = Console.ReadLine();
14            Console.WriteLine("Enter your description:");
15            desc = Console.ReadLine();
16            player = new Player(name, desc);
17
18            Item sword= new Item(new string[] { "pistol"}, "50 cal pistol", "This is
↪ a 50cal pistol");
19            Item stool = new Item(new string[] { "stool" }, "50cm stool", "This is a
↪ stool");
20            Item gem = new Item(new string[] { "gem" }, "a gem", "a bright red
↪ crystal");
21            Bag bag = new Bag(new string[] { "bag" }, "bag", "This is a expensive
↪ bag");
22
23            Location location = new Location("a conflict", "In world");
24            Location ambulance = new Location("ambulance", "In conflict");
25
26            Path ambulancenorth =new Path(new string[] { "north","n" }, "north",
↪ "ambulance north", location );
27            Path locationeast = new Path(new string[] { "east", "e" }, "east",
↪ "ambulanceineast", ambulance);
28            location.AddPath(locationeast);
29            ambulance.AddPath(ambulancenorth);
30
31
32
33
34            player.Inventory.Put(stool);
35            player.Inventory.Put(bag);
36            player.Inventory.Put(sword);
37            bag.Inventory.Put(gem);
38
39            // error solved below
40            player.Location = location;
41
42            location.Inventory.Put(sword);
43            location.Inventory.Put(stool);
44            location.Inventory.Put(bag);
45
46
47            // command loop
```

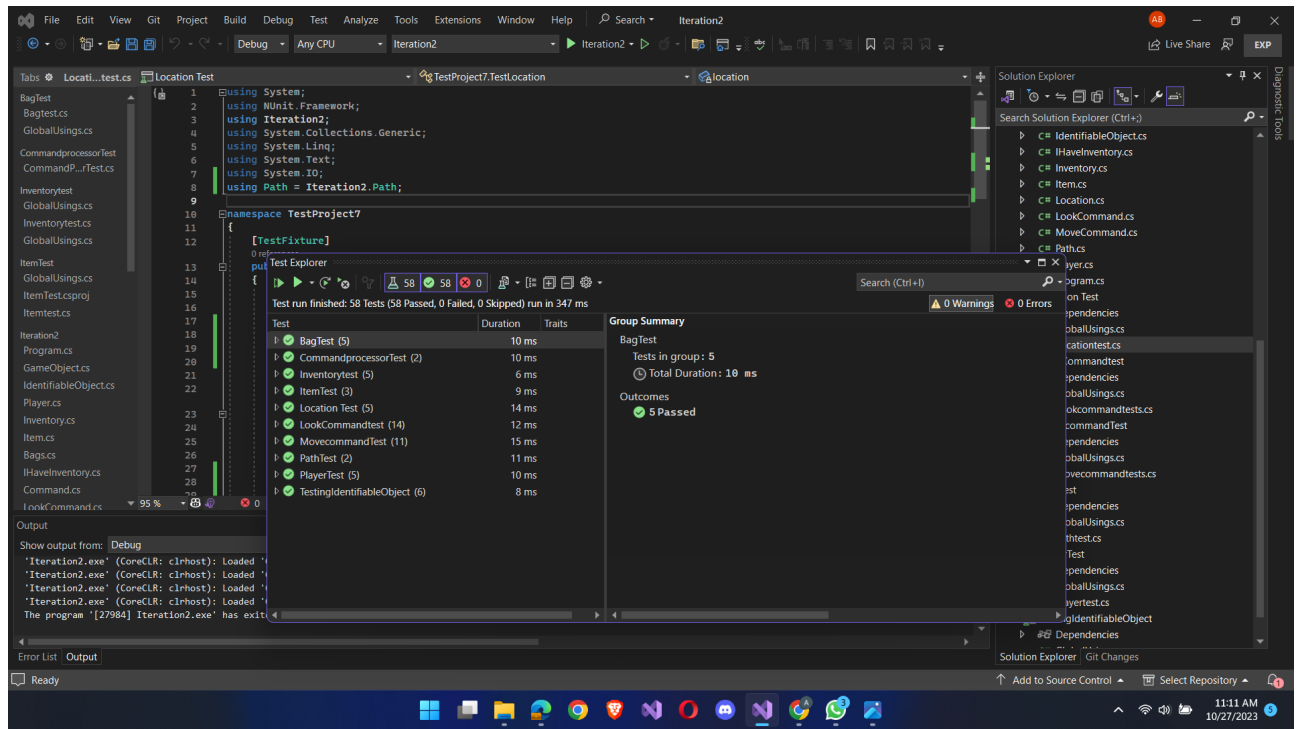
```
48
49     bool quit = false;
50     string cmd;
51     string cmdInArray;
52
53     CommandProcessor command = new CommandProcessor();
54
55     LookCommand look = new LookCommand();
56     MoveCommand move = new MoveCommand();
57
58
59     while (!quit)
60     {
61         Console.WriteLine("\nEnter a command:");
62         cmd = Console.ReadLine().ToLower();
63         if (cmd == "quit")
64         {
65             quit = true;
66         }
67         else
68         {
69             Console.WriteLine(command.Execute(player, cmd.Split()));
70         }
71     }
72
73 }
74
75 }
76 }
```

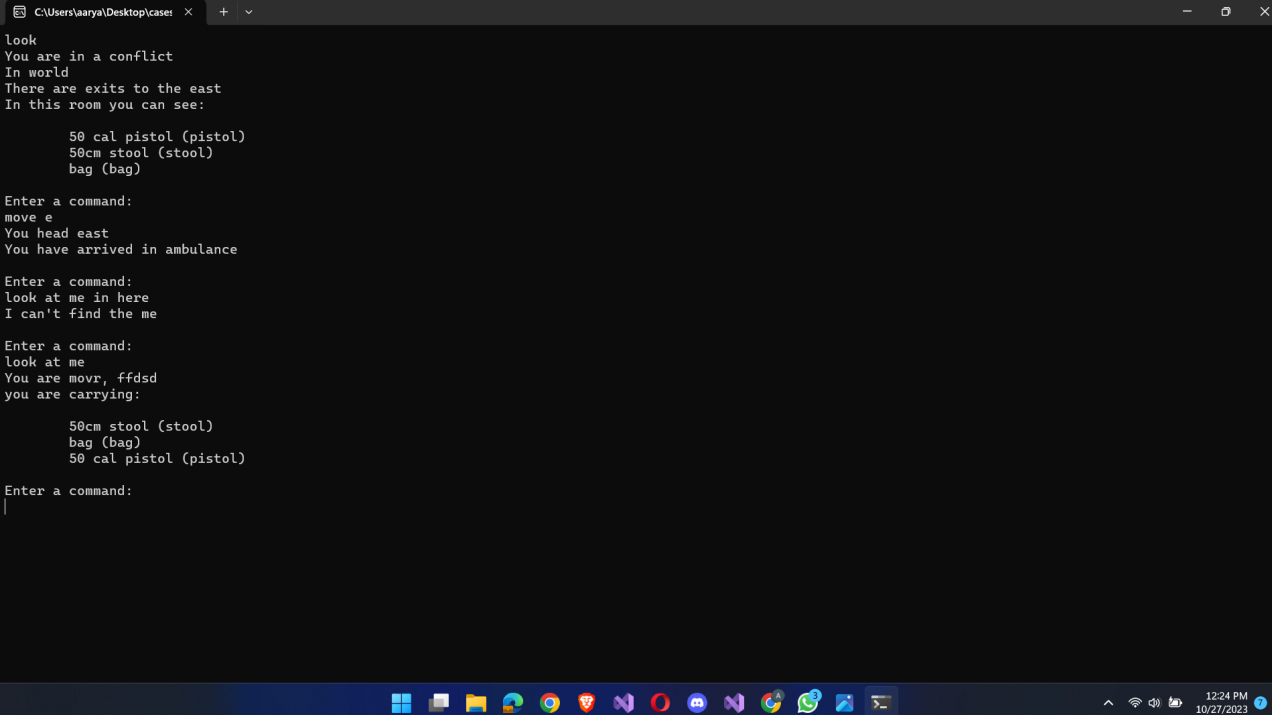
```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Iteration2
8  {
9      public class CommandProcessor : Command
10     {
11         private List<Command> _commands;
12
13         public CommandProcessor() : base(new string[] { "command" })
14         {
15             _commands = new List<Command>
16             {
17                 new LookCommand(),
18                 new MoveCommand()
19             };
20         }
21         public override string Execute(Player p, string[] text)
22         {
23             foreach (Command cmd in _commands)
24             {
25                 if (cmd.AreYou(text[0]))
26                 {
27                     return cmd.Execute(p, text);
28                 }
29             }
30             return "Error in command input";
31         }
32     }
33 }
```

```
1
2
3
4 using Iteration2;
5 using Path = Iteration2.Path;
6
7 namespace TestProject10
8 {
9
10     [TestFixture]
11     public class CommandProcessorTest
12     {
13         CommandProcessor command = new CommandProcessor();
14         Player player;
15         Item gem;
16         Location location;
17         Location destination;
18         Path path;
19         [SetUp]
20         public void SetUp()
21         {
22             location = new Location("a conflict", "In World");
23             player = new Player("Aaryan", "the student");
24             destination = new Location("ambulance", "In conflict");
25             path = new Path(new string[] { "east" }, "east", "this is east",
↪ destination);
26             gem = new(new string[] { "gem" }, "a gem", "a bright red crystal");
27             player.Location = location;
28             location.AddPath(path);
29             player.Inventory.Put(gem);
30         }
31
32         [Test]
33         public void TestExecuteMove()
34         {
35             Assert.That(player.Location, Is.SameAs(location));
36             command.Execute(player, new string[] { "move", "east" });
37             Assert.That(player.Location, Is.SameAs(destination));
38         }
39
40         [Test]
41         public void TestExecuteLook()
42         {
43             string actual = command.Execute(player, new string[] { "look", "at",
↪ "gem" });
44             string expected = "a bright red crystal";
45             Assert.That(actual, Is.EqualTo(expected));
46         }
47     }
48 }
49
50
51 }
```









A screenshot of a Windows terminal window with a dark background. The window title bar shows the path "C:\Users\vaarya\Desktop\cases" and standard window controls. The terminal displays the following text:

```
look
You are in a conflict
In world
There are exits to the east
In this room you can see:
    50 cal pistol (pistol)
    50cm stool (stool)
    bag (bag)

Enter a command:
move e
You head east
You have arrived in ambulance

Enter a command:
look at me in here
I can't find the me

Enter a command:
look at me
You are movr, ffdsd
you are carrying:
    50cm stool (stool)
    bag (bag)
    50 cal pistol (pistol)

Enter a command:
|
```

The Windows taskbar is visible at the bottom, showing various application icons and the system clock indicating 12:24 PM on 10/27/2023.