Swinburne University of Technology

COS20007 Object Oriented Programming

Case Study - Iteration 8 - Command Processor

PDF generated at 18:11 on Tuesday 23rd May, 2023

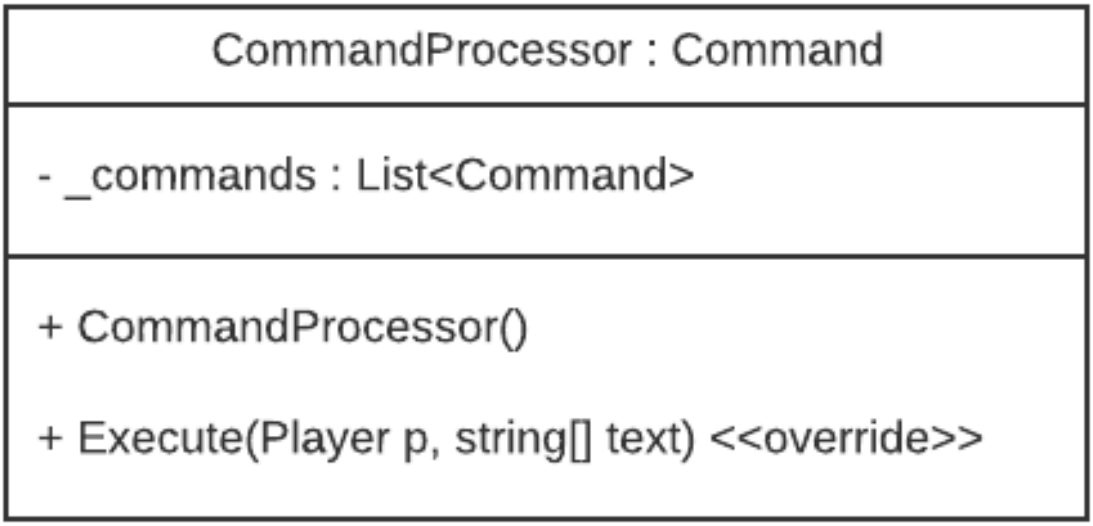
```
namespace SwinAdventure
{
    class MainClass
    {

        public static void Main(string[] args)
        {
            //local variables
            string name;
            string desc;
            Player player;

            //setting up player
            Console.WriteLine("Enter player name:");
            name = Console.ReadLine();
            Console.WriteLine("Enter player description:");
            desc = Console.ReadLine();

            player = new Player(name, desc);

            //setting up items and inventory
            Item sword = new Item(new string[] { "Sword" }, "a bronze sword", "This is a bronze sword");
            Item bat = new Item(new string[] { "Bat" }, "a hard bat", "This is a hard bat");
            Item gem = new Item(new string[] { "gem" }, "a gem", "a bright red crystal");

            Bag bag = new Bag(new string[] { "bag" }, "bag", "This is a good bag");

            player.Inventory.Put(sword);
            player.Inventory.Put(bat);
            player.Inventory.Put(bag);
            bag.Inventory.Put(gem);

            // setting up location
            Item key = new Item(new string[] { "key" }, "a key", "This is a key");
            Location hallway = new Location("the hallway", "This is the main hallway");
            Location garden = new Location("the garden", "This is a garden");

            Path hallwaySouth = new Path(new string[] { "south", "s" }, "south", "kold", garden);
            hallway.AddPath(hallwaySouth);


            hallway.Inventory.Put(key);
            player.Location = hallway;

            // command loop
            bool quit = false;
            string cmd;
            string[] cmdInArray;
```

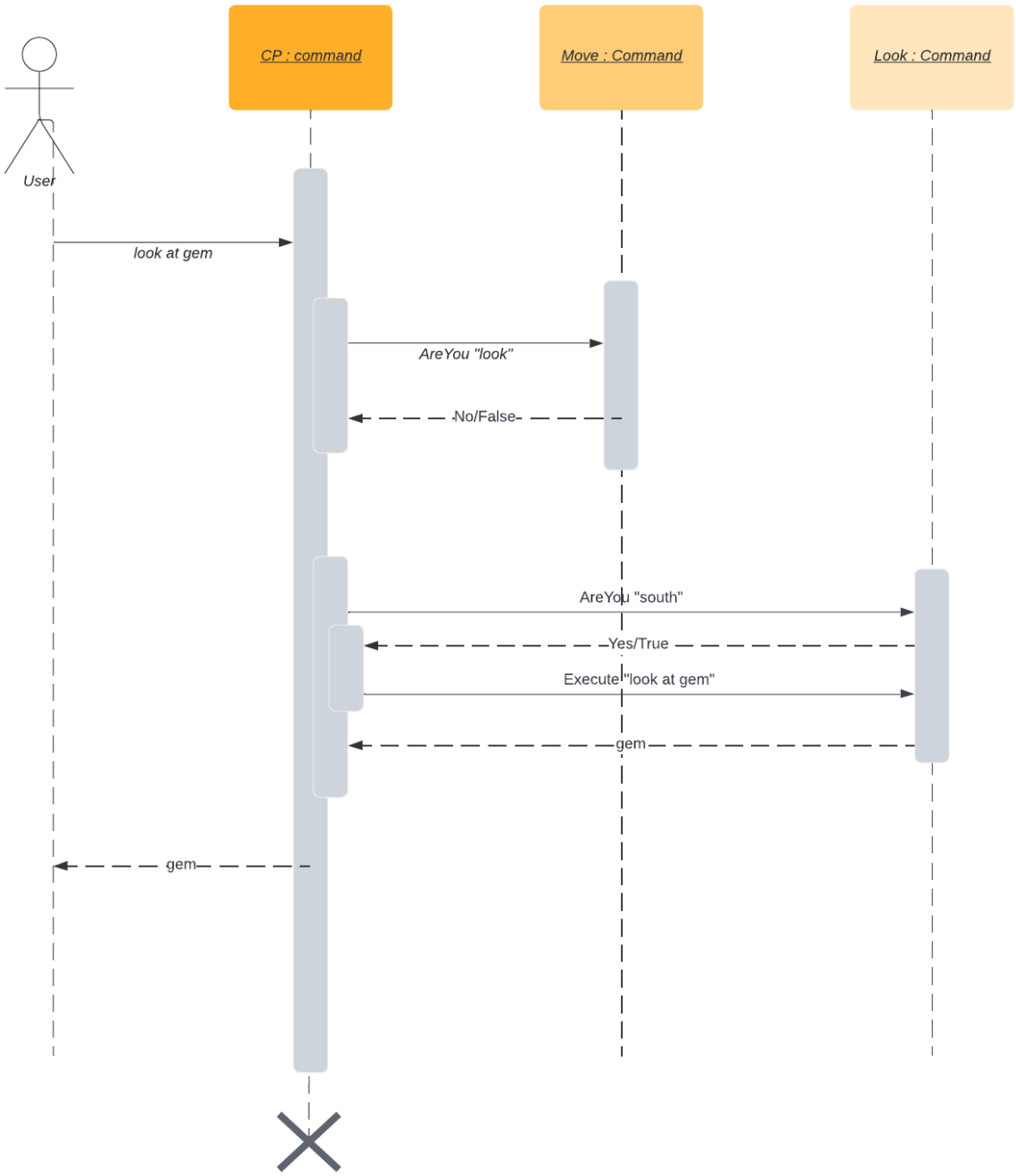
```
            CommandProcessor command = new CommandProcessor();

            while (!quit)
            {
                Console.WriteLine("\nCommand:");
                cmd = Console.ReadLine().ToLower();
                cmdInArray = cmd.Split();

                if (cmd == "quit")
                {
                    quit = true;
                }
                else
                {
                    Console.WriteLine(command.Execute(player, cmdInArray));
                }
            }
        }
    }
}
```

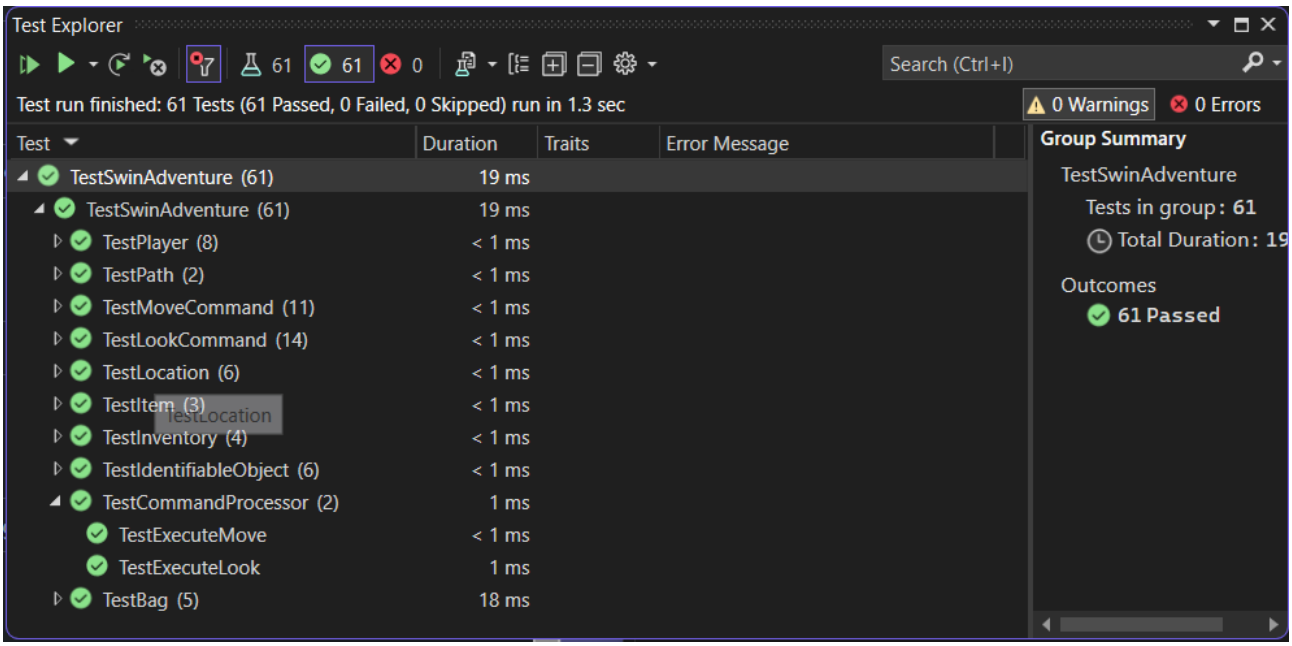
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SwinAdventure
{
    public class CommandProcessor : Command
    {
        private List<Command> _commands;

        public CommandProcessor() : base(new string[] { "command" })
        {
            _commands = new List<Command>
            {
                new LookCommand(),
                new MoveCommand()
            };
        }

        public override string Execute(Player p, string[] text)
        {
            foreach (Command cmd in _commands)
            {
                if (cmd.AreYou(text[0]))
                {
                    return cmd.Execute(p, text);
                }
            }
            return "Error in command input.";
        }
    }
}
```

```
using NUnit.Framework.Internal;
using Path = SwinAdventure.Path;

namespace TestSwinAdventure
{
    [TestFixture]
    public class TestCommandProcessor
    {
        CommandProcessor command = new CommandProcessor();
        Player player;
        //for look command
        Item gem;

        // for move command
        Location location;
        Location destination;
        Path path;

        [SetUp]
        public void SetUp()
        {
            player = new Player("shah", "the student");


            gem = new Item(new string[] { "gem" }, "a gem", "a bright red crystal");
            player.Inventory.Put(gem);

            location = new Location("a garden", "This is a garden");
            destination = new Location("a house", "This is a house");
            path = new Path(new string[] { "south" }, "south", "this is south", destination);

            player.Location = location;
            location.AddPath(path);
        }

        [Test]
        public void TestExecuteLook()
        {
            string actual = command.Execute(player, new string[] { "look", "at", "gem" });
            string expected = "a bright red crystal";

            Assert.That(actual, Is.EqualTo(expected));
        }

        [Test]
        public void TestExecuteMove()
        {
            Assert.That(player.Location, Is.SameAs(location));
            command.Execute(player, new string[] { "move", "south" });
            Assert.That(player.Location, Is.SameAs(destination));
        }
```

```


    }
}
```


CommandProcessor : Command
- _commands : List<Command>
+ CommandProcessor()
+ Execute(Player p, string[] text) <<override>>


CP : command
Move : Command
Look : Command
User
look at gem
AreYou "look"
No/False
AreYou "south"
Yes/True
Execute "look at gem"
gem
gem


Test Explorer
61
61
0
Search (Ctrl+I)
Test run finished: 61 Tests (61 Passed, 0 Failed, 0 Skipped) run in 1.3 sec
0 Warnings
0 Errors
Test
Duration
Traits
Error Message
TestSwinAdventure (61) 19 ms
TestSwinAdventure (61) 19 ms
TestPlayer (8) < 1 ms
TestPath (2) < 1 ms
TestMoveCommand (11) < 1 ms
TestLookCommand (14) < 1 ms
TestLocation (6) < 1 ms
TestItem (3) < 1 ms
TestInventory (4) < 1 ms
TestIdentifiableObject (6) < 1 ms
TestCommandProcessor (2) 1 ms
TestExecuteMove < 1 ms
TestExecuteLook 1 ms
TestBag (5) 18 ms
Group Summary
TestSwinAdventure
Tests in group: 61
Total Duration: 19
Outcomes
61 Passed

```
C:\Users\shahn\source\repos\

Enter player name:
shah
Enter player description:
sj

Command:
look
You are in the hallway
This is the main hallway
There are exists to the south.
In this room you can see:
a key (key)


Command:
move south
You head south
You have arrived in the garden

Command:
```