Swinburne University of Technology

COS20007 Object Oriented Programming

Case Study - Iteration 8 - Command Processor

PDF generated at 20:21 on Tuesday 21st November, 2023

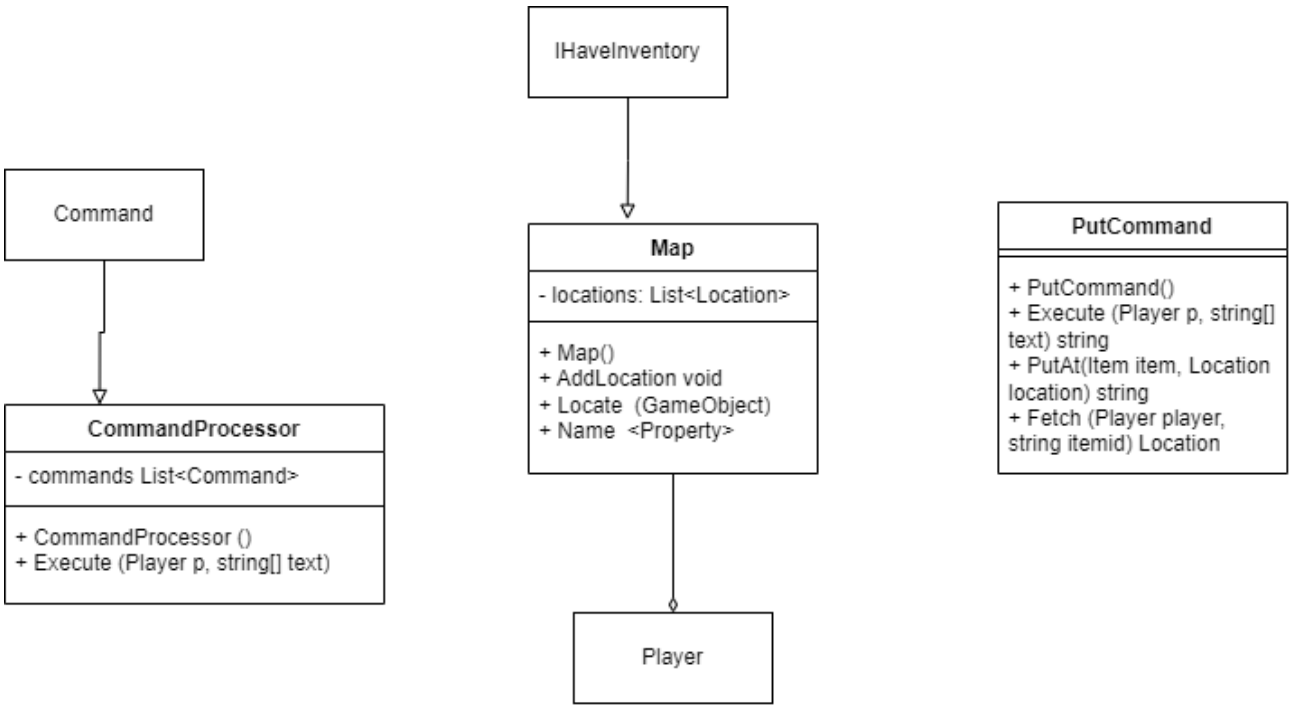
```
using System;
using System.Net.Http.Headers;

namespace Ass24
{
    public class Run
    {
        public static void Main(string[] args)
        {
            //Get info
            Console.WriteLine("Enter your name:");
            string name = Console.ReadLine();

            Console.WriteLine("Enter your description:");
            string desc = Console.ReadLine();

            Console.WriteLine($"You are {name},{desc}...");

            Player player = new Player(name, desc);

            Item item1 = new Item(new string[] { "white", "flower" }, "flower", "a whiteflower");
            Item item2 = new Item(new string[] { "dark", "chocolate" }, "chocolate", "a darkchocolate");

            player.Inventory.Put(item1);
            player.Inventory.Put(item2);
            Bag bag = new Bag(new string[] { "bagid1", "bagid2" }, name, desc);
            Item item3 = new Item(new string[] { "black", "spoon" }, "blackspoon", "a blackspoon");
            bag.Inventory.Put(item3);
            player.Inventory.Put(bag);

            Location _loc1 = new Location(new string[] { "location", "bluelock" }, "BlueLock", "a sports stadium");
            Location swin = new Location(new string[] { "location", "swinburne" }, "Swinburne", "a school");
            Location _loc3 = new Location(new string[] { "location", "graden" }, "Garden", "a garden");

            player.Location = swin;


            Path _path1 = new Path("out", _loc1, swin);
            Path _path2 = new Path("in", swin, _loc1);
            swin.AddPath(_path1);
            swin.AddPath(_path2);
            swin.Inventory.Put(item1);

            Console.WriteLine("Welcome to Swin Adventure! You have arrived in the Hallway Command");
            //Process
            string input;
```

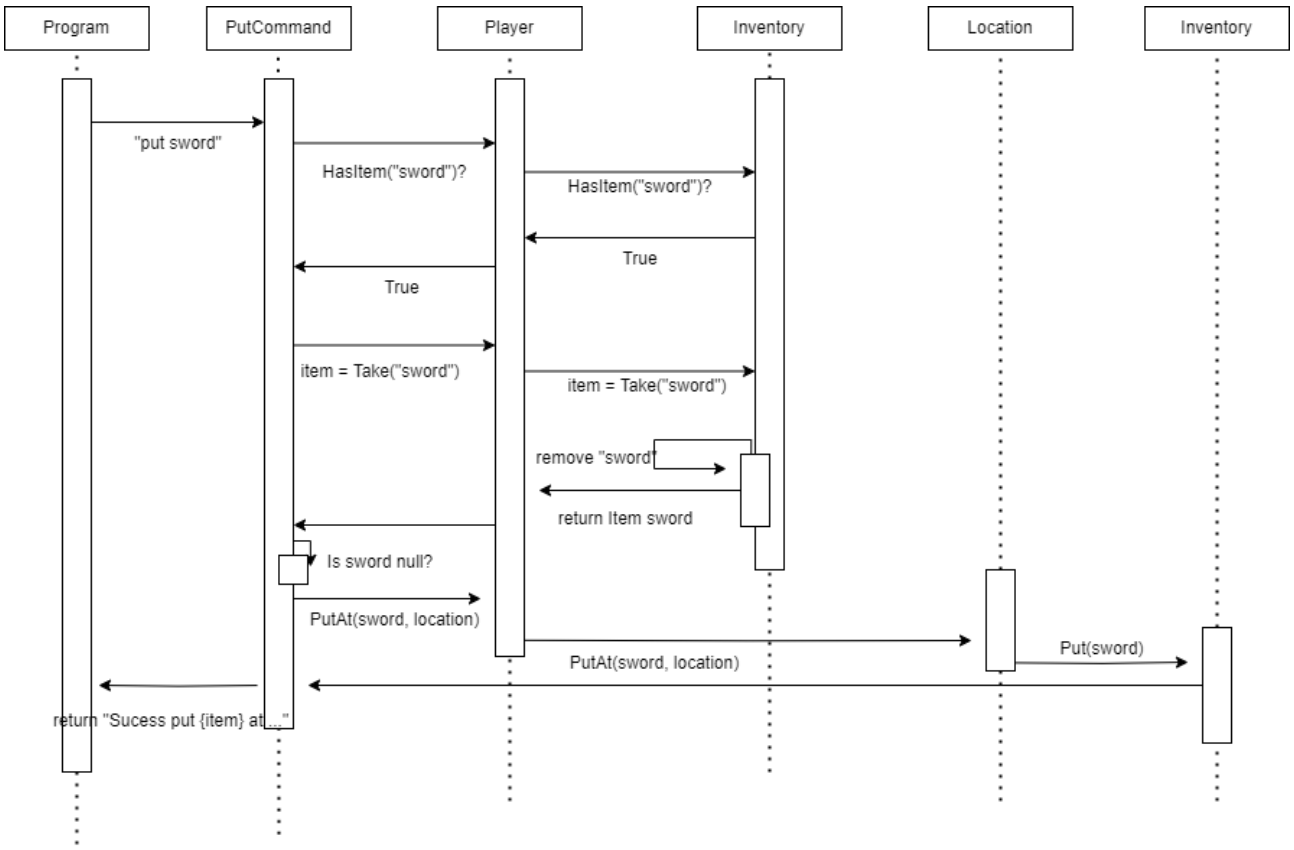
```
47              Command c = new CommandProcessor();
48              //Console.WriteLine(c.Execute(player, new string[] { "look" }));
49
50              while (true)
51              {
52                  Console.Write("Enter command: ");
53                  input = Console.ReadLine();
54
55                  if (input.ToLower() != "quit")
56                  {
57                      string[] inputArray = input.Split(' ');
58                      Console.WriteLine(c.Execute(player, inputArray));
59                  }
60                  else
61                  {
62                      Console.WriteLine("Bye");
63                      Console.ReadKey();
64                      break;
65                  }
66              }
67
68
69          }
70      }
71  }
```

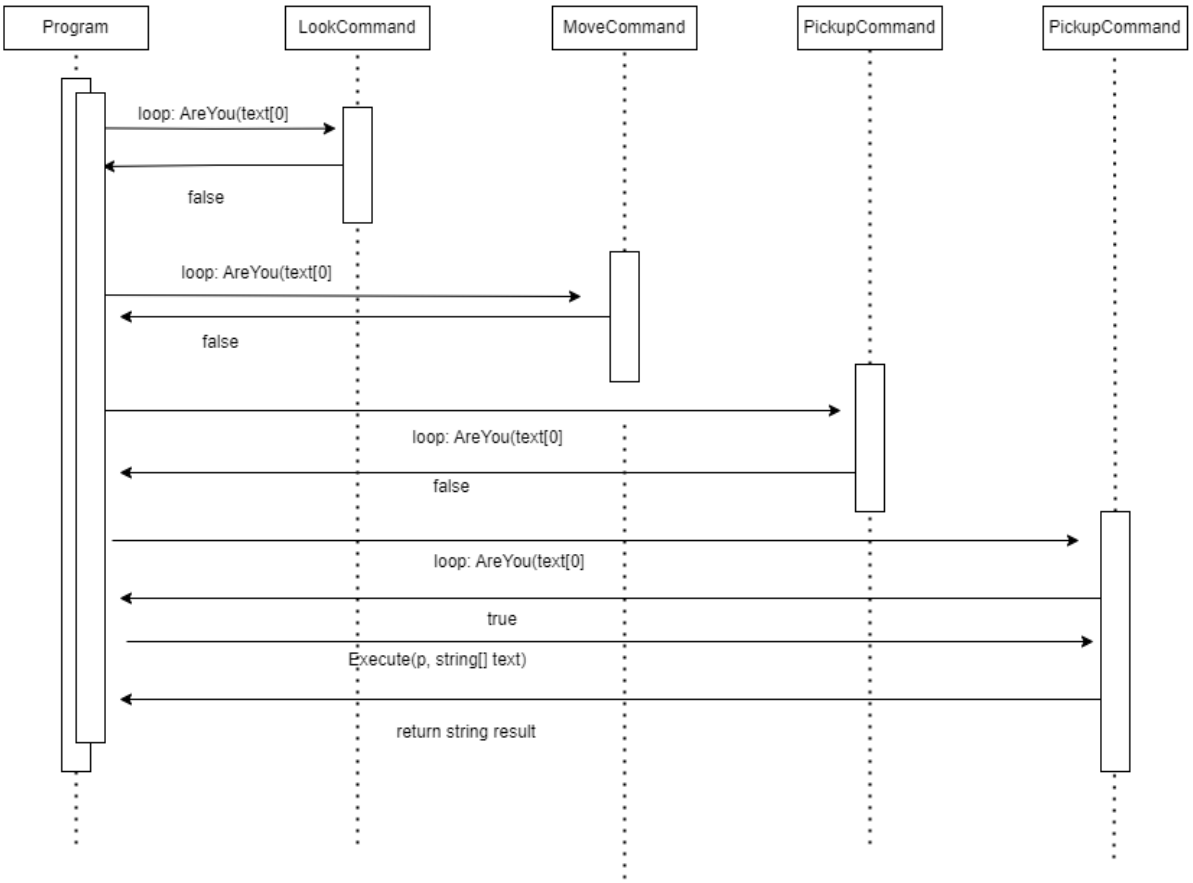
```
using Microsoft.VisualStudio.TestPlatform.Utilities;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Input;

namespace Ass24
{
    public class CommandProcessor : Command
    {
        List<Command> _commands;

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

        public override string Execute(Player p, string[] text)
        {
            foreach (Command cmd in _commands)
            {
                if (cmd.AreYou(text[0].ToLower()))
                {
                    return cmd.Execute(p, text);
                }
            }
            return $"I don't understand {string.Join("",text)}";

        }
    }
}
```

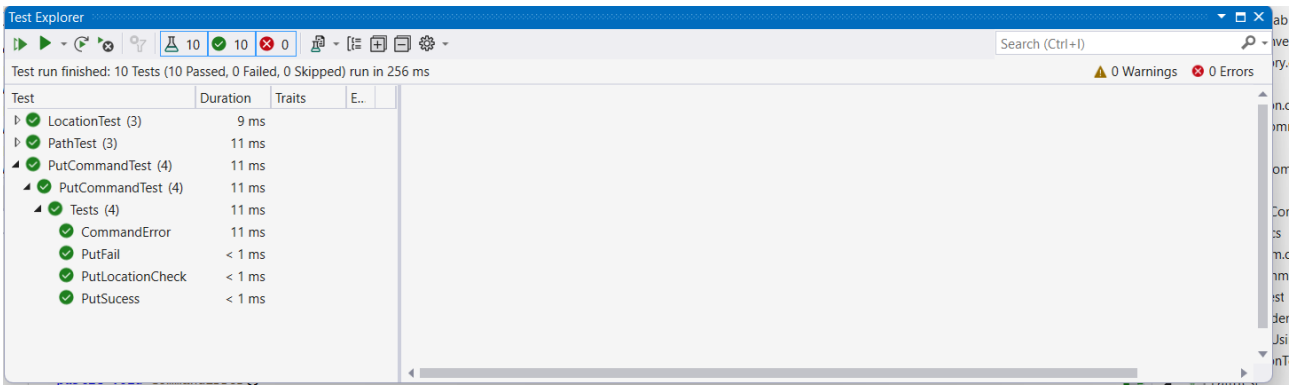
```
using System.Numerics;
using Ass24;

namespace PutCommandTest
{
    public class Tests
    {

        Location _location1;
        Location _location2;
        Player player;
        CommandProcessor _command;
        Item sword;


        [SetUp]
        public void Setup()
        {
            _command = new CommandProcessor();
            player = new Player("lily", "Tired");

            _location1 = new Location(new string[] { "swin", "burne" }, "Swinburne", "A place to study.");
            _location2 = new Location(new string[] { "home", "house" }, "Home", "A place to sleep.");
            sword = new Item(new string[] { "sword" }, "Sword", " sturdy weapon.");
            player.Location = _location1;

            player.Map.AddLocation(_location1);
            player.Map.AddLocation(_location2);

            player.Inventory.Put(sword);
        }

        [Test]
        public void CommandError()
        {
            Assert.AreEqual(_command.Execute(player, new string[] { "wrong", "sword" }), "I don't understand wrongsword");
        }

        [Test]
        public void PutSucess()
        {
            Assert.AreEqual(_command.Execute(player, new string[] { "put","sword"}), "Sucess put Sword in Swinburne");
        }

        [Test]
        public void PutLocationCheck()
        {
            _command.Execute(player, new string[] { "put", "sword" });
            Assert.AreEqual(_location1.Locate("sword").Name, "Sword");
```

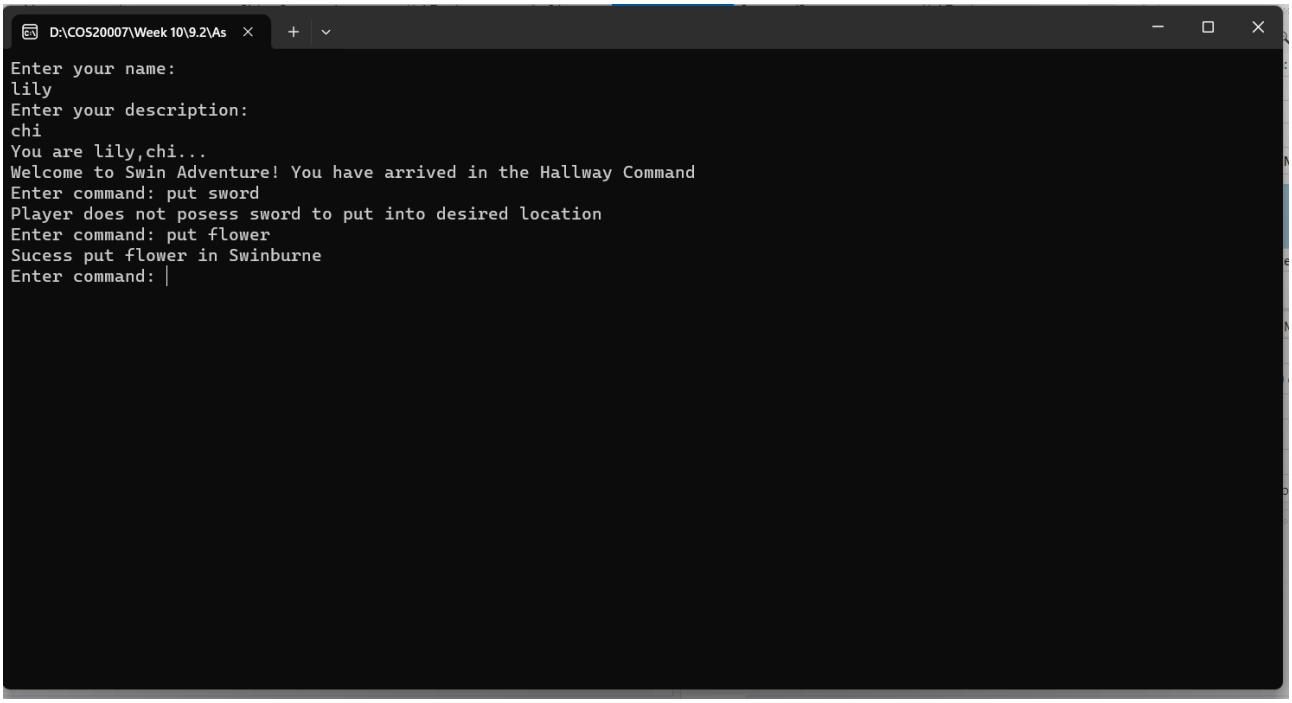
```
50          }
51
52          [Test]
53          public void PutFail()
54          {
55              Assert.AreEqual(_command.Execute(player, new string[] { "put", "pen" }),
    ↪  "Player does not posess pen to put into desired location");
56          }
57      }
58  }
```


IHaveInventory
Command
Map
- locations: List<Location>
+ Map()
+ AddLocation void
+ Locate (GameObject)
+ Name <Property>
PutCommand
+ PutCommand()
+ Execute (Player p, string[] text) string
+ PutAt(Item item, Location location) string
+ Fetch (Player player, string itemid) Location
CommandProcessor
- commands List<Command>
+ CommandProcessor ()
+ Execute (Player p, string[] text)
Player


Program
PutCommand
Player
Inventory
Location
Inventory
"put sword"
HasItem("sword")?
HasItem("sword")?
True
True
item = Take("sword")
item = Take("sword")
remove "sword"
return Item sword
Is sword null?
PutAt(sword, location)
Put(sword)
PutAt(sword, location)
return "Sucess put {item} at ..."


Program
LookCommand
MoveCommand
PickupCommand
PickupCommand
loop: AreYou(text[0]
false
loop: AreYou(text[0]
false
loop: AreYou(text[0]
false
loop: AreYou(text[0]
true
Execute(p, string[] text)
return string result


Test Explorer
10
10
0
Search (Ctrl+I)
Test run finished: 10 Tests (10 Passed, 0 Failed, 0 Skipped) run in 256 ms
0 Warnings
0 Errors
Test
Duration
Traits
E..
LocationTest (3) 9 ms
PathTest (3) 11 ms
PutCommandTest (4) 11 ms
PutCommandTest (4) 11 ms
Tests (4) 11 ms
CommandError 11 ms
PutFail < 1 ms
PutLocationCheck < 1 ms
PutSucess < 1 ms


D:\COS20007\Week 10\9.2\As
Enter your name:
lily
Enter your description:
chi
You are lily,chi...
Welcome to Swin Adventure! You have arrived in the Hallway Command
Enter command: put sword
Player does not posess sword to put into desired location
Enter command: put flower
Sucess put flower in Swinburne
Enter command: