

**COE312 Software Design for Engineers**

**Homework 3: Design Patterns**

**Academic Integrity Pledge**

|  |
| --- |
| As a student of American University of Sharjah, I here by state that I will abide by the AUS Integrity Pledge that:   * I will hold myself accountable for all that I say and write. * I will hold myself responsible for the academic integrity of my work * I will not carry out unauthorized copying or printing of the work of others * I will not misrepresent my work nor give or receive unauthorized aid * I will behave in a manner that demonstrates concern for the personal dignity, rights and freedoms of all members of the community * I will respect university property and the property of others; and * I will not tolerate a lack of respect for these values.   **Student Name:Mohamed Jassim Alobeidilli**  **Student ID:75629**  **Student Name:Mohammed Walid Khan**  **Student ID:79491** |

**Note: You must submit the code files as well.**

1. **State how the software design patterns can be used to implement the game.**

We have subject generating data (Watch and Character), and we have observers that are wanting this data and they may use it as they want (Like Hermione). Hence we need to make use of the Observer Pattern, and we only need to have one instance, so Singleton is used.

Also, because we have like choices of different spells we need to execute, we need something like a remote that we can choose from a Menu of actions. So we will have a Control Panel that will have these commands. Hence, this is why we need the Command Pattern.

Also, we have a recipe that needs to be enforced to make Potions, we have Invisibility Potion and Polyjuice Potion, and these potions will have some steps common and some unique specific to that potion. Hence, we need to make sure we use the Template Method pattern to enforce this recipe.

1. **Implement the game using the software design patterns and threads.**

* The Java code

*Please add subtitles with the class names. Paste only the new code, and* ***not*** *screenshots of the code.*

**Subject**

**package** HW3;

**public** **interface** Subject {

**void** RegisterObserver(Observer o);

**void** RemoveObserver(Observer o);

**void** NotifyObserver();

}

**ConcreteSubject**

**package** HW3;

**import** java.util.\*;

**public** **class** ConcreteSubject **implements** Subject

{

**private** ArrayList<Observer> Observers;

**public** ConcreteSubject()

{

Observers = **new** ArrayList<Observer>();

}

@Override

**public** **void** RegisterObserver(Observer o)

{

Observers.add(o);

}

@Override

**public** **void** RemoveObserver(Observer o)

{

**int** i = Observers.indexOf(o);

**if** (i >= 0) Observers.remove(i);

}

@Override

**public** **void** NotifyObserver()

{

**for** (**int** i = 0; i < Observers.size(); i++)

{

Observer observer = (Observer) Observers.get(i);

observer.update();

}

}

}

**Observer**

**package** HW3;

**public** **interface** Observer {

**void** update();

}

**ConcreteObserver**

**package** HW3;

**public** **class** ConcreteObserver **implements** Observer

{

**private** Subject subject;

**public** ConcreteObserver(Subject subject)

{

**this**.subject = subject;

subject.RegisterObserver(**this**);

}

@Override

**public** **void** update()

{

}

}

**Watch**

**package** HW3;

**public** **class** Watch **extends** ConcreteSubject **implements** Runnable {

**boolean** b;

**private** **static** Watch *instance*;

Thread t;

**private** Watch()

{

**super**();

t = **new** Thread(**this**);

t.start();

}

**public** **static** **synchronized** Watch getInstance()

{

**if**(*instance* == **null**)

{

*instance* = **new** Watch();

}

**return** *instance*;

}

@Override

**public** **void** run()

{

**try**

{

**for** (**int** i = 0; i < 5; i++)

{

t.*sleep*(60000);

System.***out***.println((i+1)+" minutes have passed, you have "+(5-(i+1))+" minutes left..");

}

System.***out***.println("You have 10 seconds to close the map before you get caught");

b = **true**;

**try**

{

t.*sleep*(10000);

}

**catch** (InterruptedException e)

{

e.getStackTrace();

}

System.***out***.println("you have been caught using the map and the map has been confiscated and you lost");

System.*exit*(1);

}

**catch** (Exception e)

{

e.getStackTrace();

}

}

}

**Character**

**package** HW3;

**import** java.util.Scanner;

**public** **class** Character **extends** ConcreteSubject **implements** Runnable {

String loc;

Watch w;

Command[] spells;

Spells s;

ControlPanel cp;

Thread t;

**boolean** polyPotion = **false**;

**boolean** invisibilePotion = **false**;

**private** **static** Character *ch*;

**private** Character(Watch w, Spells s) {

**super**();

**this**.loc = "Entrance";

**this**.w = w;

**this**.s = s;

Expelliarmus expelliarmus = **new** Expelliarmus(s);

Stupefy sptupefy = **new** Stupefy(s);

Protego protego = **new** Protego(s);

ExpectoPatronum expectoPatronum = **new** ExpectoPatronum(s);

Command[] sp = { expelliarmus, sptupefy,protego, expectoPatronum };

spells = sp;

cp = **new** ControlPanel(spells);

System.***out***.println("The Entrance"

+ "\n\nYou arrive at the doors of Hogwarts. The door on the north wall leads to the dining\r\n" +

"hall, the door to the east leads to the Potions class, and the door to the west leads to\r\n" +

"the Library.");

**boolean** status = **false**;

String exit = "mischief managed";

Scanner object = **new** Scanner(System.***in***);

String c = object.nextLine();

**while** (!status) {

**if** (c.contains("Potions") || c.contains("East")) {

setLocation("Potions");

**new** PortionsClassroom(**this**);

status = **true**;

}

**else** **if** (c.contains("Library") || c.contains("West")) {

setLocation("Library");

**new** Library(**this**);

status = **true**;

}

**else** **if** (c.contains("Dining") || c.contains("North")) {

setLocation("Dining");

**new** DiningHall(**this**);

status = **true**;

} **else** **if** (c.equals(exit)) {

status = **true**;

} **else** {

System.***out***.println("Invalid option, try again:");

System.***out***.println();

c = object.nextLine();

}

t = **new** Thread(**this**);

t.start();

}

}

**public** **static** **synchronized** Character getInstance(Watch watch, Spells spell) {

**if** (*ch* == **null**) {

*ch* = **new** Character(watch, spell);

}

**return** *ch*;

}

**public** **synchronized** String getLocation() {

**return** loc;

}

**public** **synchronized** **void** setLocation(String loc) {

**this**.loc = loc;

}

@Override

**public** **void** run() {

Scanner object = **new** Scanner(System.***in***);

**while** (**true**) {

**if** (w.b) {

String chief = object.nextLine();

**if** (chief.equals("mischief managed")) {

System.***out***.println("You closed the map and did not manage to find Hermione");

System.*exit*(1);

}

NotifyObserver();

}

}

}

}

**Hermione**

**package** HW3;

**public** **class** Hermione **extends** ConcreteObserver

{

Character ch;

**public** Hermione(Character ch)

{

**super**(ch);

**this**.ch = ch;

ch.RegisterObserver(**this**);

}

@Override

**public** **void** update()

{

**if** (ch.getLocation().equals("Library"))

{

System.***out***.println("Hermione: I am here!");

ch.RemoveObserver(**this**);

}

}

}

**ControlPanel**

**package** HW3;

**public** **class** ControlPanel

{

Command [] slots;

**public** ControlPanel(Command[] slots)

{

**this**.slots = slots;

}

**public** **void** castingSpell(**int** index, String deathEaters)

{

slots[index].execute(deathEaters);

}

}

**Command**

**package** HW3;

**public** **interface** Command

{

**void** execute(String deathEater);

}

**ExpectoPatronum**

**package** HW3;

**public** **class** ExpectoPatronum **implements** Command

{

Spells s;

ExpectoPatronum(Spells s)

{

**this**.s = s;

}

@Override

**public** **void** execute(String deathEater)

{

s.ExpectoPatronum(deathEater);

}

}

**Expelliarmus**

**package** HW3;

**public** **class** Expelliarmus **implements** Command

{

Spells s;

Expelliarmus(Spells s)

{

**this**.s = s;

}

@Override

**public** **void** execute(String deathEater )

{

s.Expelliarmus(deathEater );

}

}

**Protego**

**package** HW3;

**public** **class** Protego **implements** Command {

Spells s;

Protego(Spells s) {

**this**.s = s;

}

@Override

**public** **void** execute(String deathEater) {

s.Protego(deathEater);

}

}

**Stupefy**

**package** HW3;

**public** **class** Stupefy **implements** Command{

Spells s;

Stupefy(Spells s)

{

**this**.s = s;

}

@Override

**public** **void** execute(String deathEater)

{

s.Stupefy(deathEater);

}

}

**Spells**

**package** HW3;

**public** **class** Spells {

**void** Expelliarmus(String deathEater) {

System.***out***.println("You defeated " + deathEater + "!... the weapon has been disarmed from your enemy");

}

**void** Stupefy(String deathEater) {

System.***out***.println("The " + deathEater + " has been stunned");

}

**void** Protego(String deathEater) {

System.***out***.println("You saved yourself from " + deathEater

+ " ...now you can now use the cauldron and ingredients to make the\r\n"

+ "Polyjuice potion or the invisibility potion to proceed without anyone recognizing you…");

}

**void** ExpectoPatronum(String deathEater) {

System.***out***.println(

"You are now protected from Dark Magic. The " + deathEater + " can't attack you now");

}

}

**Potions**

**package** HW3;

**public** **abstract** **class** Potions {

**final** **void** preparePotion(String add, String str,String temp)

{

add(add);

stir(str);

heat(temp);

wave();

}

**void** add(String add)

{

System.***out***.println("Adding "+add+" in cauldron");

}

**void** stir(String str)

{

System.***out***.println("Stirring "+str);

}

**void** heat(String temp)

{

System.***out***.println("Heating at "+temp+" temperature");

}

**abstract** **void** wave();

}

**InvisibiltyPotion**

**package** HW3;

**public** **class** InvisibilityPotion **extends** Potions {

@Override

**void** wave() {

}

}

**PolyjuicePotion**

**package** HW3;

**public** **class** PolyjuicePotion **extends** Potions {

**void** wave() {

System.***out***.println("Waving hand to complete the brewing...");

}

}

**School**

**package** HW3;

**import** java.util.Scanner;

**public** **abstract** **class** School {

String name;

String location;

String DeathEaters;

Character ch;

**public** School()

{

}

**public** School(String n, String l, String de, Character chr)

{

**this**.location = l;

**this**.name = n;

**this**.DeathEaters=de;

**this**.ch=chr;

}

**final** **void** place()

{

look();

next();

potionsProcedure();

fight();

}

**void** fight()

{

**if**(name.equals("Library") && !ch.invisibilePotion && !ch.polyPotion)

{

System.***out***.println("You do not have the two potions with you so you cannot fight Draco Malfoy\n" +

"Go to the Potions Classroom to get the potions.");

}

**else** {

Scanner obj = **new** Scanner(System.***in***);

String choice = "";

**boolean** status = **false**;

**while** (!status) {

choice = obj.nextLine();

**if** (choice.equals("Expelliarmus") )

{

status = **true**;

ch.cp.castingSpell(0, DeathEaters);

}

**else** **if** (choice.equals("Protego") )

{

status = **true**;

ch.cp.castingSpell(2, DeathEaters);

}

**else** **if** (choice.equals("Stupefy") )

{

status = **true**;

ch.cp.castingSpell(1, DeathEaters);

}

**else** **if** (choice.equals("Expecto Patronum") )

{

status = **true**;

ch.cp.castingSpell(3, DeathEaters);

}

**else** {

System.***out***.println("invalid choice");

}

}

}

}

**abstract** **void** look();

**abstract** **void** next();

**abstract** **void** potionsProcedure();

}

**DiningHall**

**package** HW3;

**import** java.util.Scanner;

**public** **class** DiningHall **extends** School

{

DiningHall(Character ch)

{

**super**("DiningHall", "North", "Bellatrix Lestrange", ch);

System.***out***.println("You have entered the dining hall");

Scanner c = **new** Scanner(System.***in***);

String str = c.nextLine();

**while** ( !( str.contains("look") ) )

{

System.***out***.println("Invalid option entered. Try again:");

str = c.nextLine();

}

place();

}

@Override

**void** look()

{

// **TODO** Auto-generated method stub

System.***out***.println("The dining hall is the biggest open area inside Hogwarts. Bellatrix Lestrange is here,\n" +

"get ready to fight!");

}

@Override

**void** next()

{

// **TODO** Auto-generated method stub

**boolean** status = **false**;

Scanner object = **new** Scanner(System.***in***);

**while** (!status)

{

String c = object.nextLine();

**if** (c.contains("potions") || c.contains("east") )

{

status = **true**;

ch.setLocation("Potions Classroom");

**new** PortionsClassroom(ch);

}

**else** **if** (c.contains("library") || c.contains("west") )

{

status = **true**;

ch.setLocation("Library");

**new** Library(ch);

}

**else**

{

System.***out***.println("Invalid option entered. Try again:");

}

}

}

@Override

**void** potionsProcedure()

{

}

}

**Library**

**package** HW3;

**import** java.util.Scanner;

**public** **class** Library **extends** School {

Hermione h;

Library(Character ch) {

**super**("Library", "West", "Draco Malfoy", ch);

System.***out***.println("You have entered the library.");

h = **new** Hermione(ch);

Scanner object = **new** Scanner(System.***in***);

String c = object.nextLine();

**while** (!(c.contains("look"))) {

System.***out***.println("Invalid option entered. Try again:");

c = object.nextLine();

}

place();

}

@Override

**void** look() {

System.***out***.println(

"Books is one of the core sources of knowledge that first year students are required to study in.\n"

+ "In the potions class there is Draco Malfoy who is destroying the library. He seems very angry\n"

+ "to see you and starts attacking you!");

System.***out***.println("Library is one of the most important places to study peacefully for students\r\n"

+ "Draco Malfoy is in the Libary and want to start attacking you!\r\n");

}

@Override

**void** next() {

**if** (!(!ch.polyPotion && !ch.invisibilePotion)) {

System.***out***.println("Hermione is found!");

Scanner object = **new** Scanner(System.***in***);

**while** (!object.nextLine().equals("mischief managed")) {

System.***out***.println("Those are not the correct words to close the map try again: ");

}

System.***out***.println("Hiding map contents... end.");

System.*exit*(1);

}

**else** {

Scanner object = **new** Scanner(System.***in***);

**boolean** status = **false**;

**while** (!status) {

String c = object.nextLine();

**if** (c.contains("dining") || c.contains("north")) {

status = **true**;

ch.setLocation("Dining Hall");

**new** DiningHall(ch);

} **else** **if** (c.contains("potions") || c.contains("east")) {

status = **true**;

ch.setLocation("Potions Classroom");

**new** PortionsClassroom(ch);

} **else** {

System.***out***.println("Invalid option entered. Try again:");

}

}

}

}

@Override

**void** potionsProcedure() {

}

}

**PortionsClassroom**

**package** HW3;

**import** java.util.Scanner;

**public** **class** PortionsClassroom **extends** School

{

Character ch;

PortionsClassroom(Character ch)

{

**super**("Potions Classroom", "East", "Professor Snape", ch);

System.***out***.println("You have entered the Potions Classroom");

**this**.ch = ch;

Scanner object = **new** Scanner(System.***in***);

String x = object.nextLine();

**while** ( !( x.contains("look") ) )

{

System.***out***.println("Invalid option entered. Try again:");

x = object.nextLine();

}

place();

}

@Override

**void** potionsProcedure()

{

int count1=0;

int count2=0;

boolean polcheck=false;

boolean invcheck=false;

System.***out***.println("You saved yourself…now you can now use the cauldron and ingredients to make the\n" +

"Polyjuice potion or the invisibility potion to proceed without anyone recognizing you…");

Scanner object = **new** Scanner(System.***in***);

**boolean** status = **false**;

**while** (!status) {

String x = object.nextLine();

**if**(x.startsWith("make"))

{

String potion = x.substring(5);

**if** (potion.equals("Polyjuice potion"))

{

**if** (ch.polyPotion == **false**)

{

if(invcheck==false){

**new** PolyjuicePotion().preparePotion("fluxweed and lacewings", "4 times clockwise", "high");

count1++;

if(count1 ==2){

ch.polyPotion = **true**;

}

}else System.out.println(“u need to make the invisibility potion”);

} **else** {

System.***out***.println("You already have Polyjuice potion");

}

} **else** **if** (potion.equals("Invisibility potion")) {

**if** (ch.invisibilePotion == **false**) {

if(polcheck ==false){

**new** InvisibilityPotion().preparePotion("cherries and spiders", "3 times anticlockwise", "high");

count2++;

if(count2==2){

ch.invisibilePotion = **true**;

}

}else System.out.println(“u need to make the polyjuise potion”);

} **else** {

System.***out***.println("You already have Invisible potion");

}

} **else** {

System.***out***.println("Invalid option try again: ");

}

} **else**

{

System.***out***.println("Invalid option try again: ");

}

**if**(ch.polyPotion && ch.invisibilePotion){

status = **true**;

}

}

}

@Override

**void** look() {

System.***out***.println("Potions is one of the core subjects that first year students are required to study. In the\n" +

"potions class there is Professor Snape who teaches the subject. He does not seem very happy\n" +

"to see you and starts attacking you!");

}

@Override

**void** next() {

Scanner object = **new** Scanner(System.***in***);

**boolean** status = **false**;

**while** (!status) {

String c = object.nextLine();

**if** (c.contains("library") || c.contains("west")) {

status = **true**;

ch.setLocation("Library");

**new** Library(ch);

} **else** **if** (c.contains("dining") || c.contains("north")) {

status = **true**;

ch.setLocation("Dining Hall");

**new** DiningHall(ch);

} **else** {

System.***out***.println("Invalid option entered. Try again:");

}

}

}

}

**Driver**

**package** HW3;

**import** java.util.Scanner;

**public** **class** Driver {

**public** **static** **void** main(String[] args)

{

// **TODO** Auto-generated method stub

System.***out***.println("Messrs Moony, Wormtail, Padfoot and Prongs Purveyors of Aids " +

"\nto magical Mischief-Makers are proud to present THE MARAUDER'S MAP" +

"\n\nSay the magic phrase to reveal the map.");

Scanner scanner = **new** Scanner(System.***in***);

**while** (!scanner.nextLine().equals("I solemnly swear that I am up for no good"))

{

System.***out***.println("Those are not the correct words try again: ");

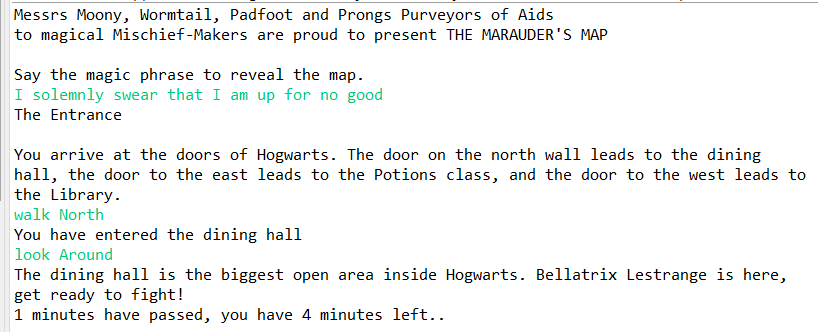
}

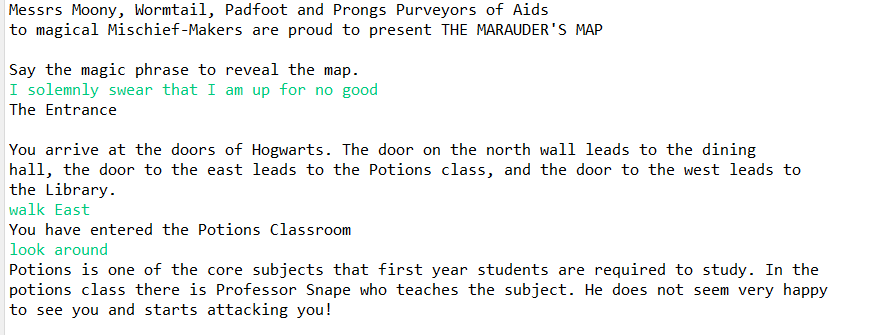
Character.*getInstance*(Watch.*getInstance*(), **new** Spells());

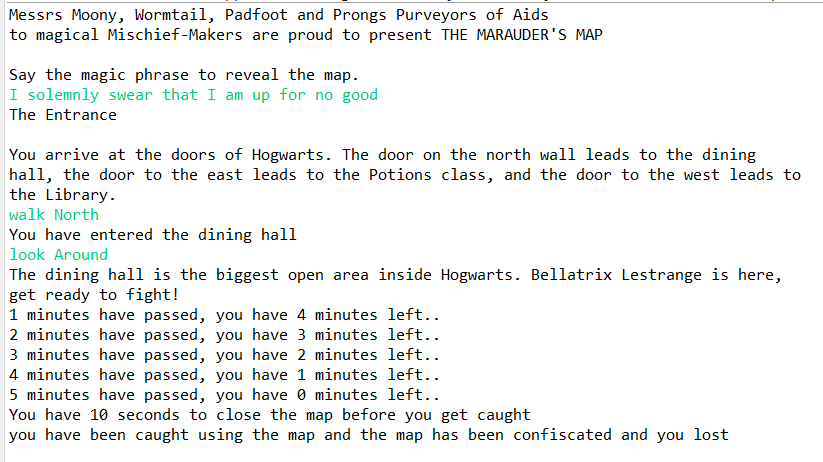
}

}

1. **Show sample outputs**
2. Paste screenshots of output showing the player in all locations, fighting all villains and making the 2 potions.
3. Paste screenshot of output showing that the player found Hermione before time is up and then hiding the map to end the game.
4. Paste screenshot of output showing that the player got caught (time is up and s/he did not hide the map).







GITHUB LINK: https://github.com/walidkhan5/COE312\_HW3.git