Jobsheet 04 - Relationships Between Classes

Erwan Majid/08/2i

Link github: https://github.com/Majid5654/Semester-3/tree/Main/JAVA%20OOP/Week%205

- **Experiment 1**

Class Processor:

Processor.java > Processor

```java
public class Processor {
    private String brand;
    private double cache;

    public Processor(){

    }

    public Processor (String brand,double cache){
        this.brand = brand;
        this.cache = cache;
    }

    public String getBrand(){
        return brand;
    }

    public void setBrand(String brand){
        this.brand = brand;
    }

    public double getCache(){
        return  cache;
    }

    public void setCache(double cache){
        this.cache = cache;
    }

    public void info(){
        System.out.printf(format:"Brand Processor = %s \n",brand);
        System.out.printf(format:"Cache Memory = %.2f\n", cache);
    }

}
```

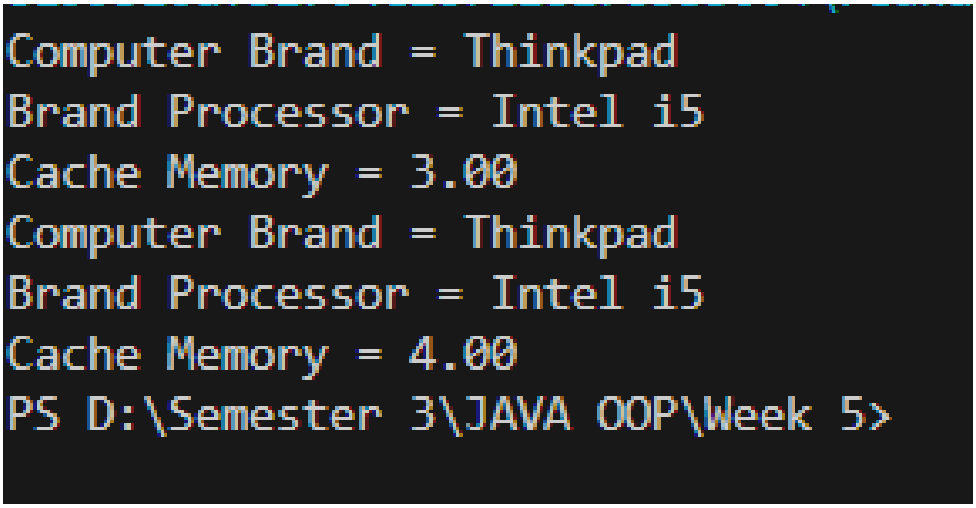
Class computer:

```java
public class Computer{
    private String brand;
    private Processor proc;

    public Computer(){

    }

    public Computer(String brand,Processor proc){
        this.brand = brand;
        this.proc = proc;
    }

    public String getBrand(){
        return brand;
    }

    public void setBrand(String brand){
        this.brand = brand;
    }

        public Processor getproc(){
            return proc;
        }

        public void setProc(Processor proc){
            this.proc = proc;
        }

    public void info(){
        System.out.println("Computer Brand = " +brand);
        proc.info();
    }
}
```

Main:

```java
public class MainExperiment1 {
    Run | Debug
    public static void main(String[] args) {

        Processor p = new Processor(brand:"Intel i5", cache:3);
        Computer L = new Computer(brand:"Thinkpad", p);

        L.info();

        Processor p1 = new Processor();
        p1.setBrand(brand:"Intel i5");
        p1.setCache(cache:4);

        Computer c1 = new Computer();
        c1.setBrand(brand:"Thinkpad");
        c1.setProc(p1);
        c1.info();



    }
}
```

Output:



Questions

Based on experiment 1, answer the related questions:

1. In the Processor class and Computer class, there are setter and getter methods for each attribute. What the purpose of method setters and getters are ?

-To protect attributes from direct access.The attributes can only be accessed and modified through getter and setter methods, allowing better control over the validation and restriction of attribute value changes.

2. Within the Processor class and Computer class, there are each a default constructor and a parameterized constructor. How is the use of the two types of constructors different ?

-Default constructor to not set any value in first inititial but it use to set value later

- parameterized constructor to set value in first initial it can reduces the chances of using uninitialized attributes.

3. Consider the Computer class, which of the 2 attributes (brand and proc), which attribute is object type ?

-proc: it because this attribute is of the type Processor, which is a class

4. Look at the Computer class, which line shows that the Computer class has a relation with the Processor class ?

- private Processor proc;

- This line establishes that the Computer class has an association with the Processor class. The attribute proc is of type Processor

5. Pay attention to the Computer class, What is the syntax of proc.info() ?

- calling a method of an object (the proc object of type Processor), which outputs the details of the processor (brand and cache memory)

6. In the MainExperiment1 class , there is a line of code:

Computer c = new Laptop("Thinkpad", p);.

- What is p ?

- p is an instance of the Processor class, initialized with the brand "Intel i5" and a cache size of 3.0.

And what happens if the line of code is changed to:

Computer c = new Laptop("Thinkpad", new Processor("Intel i5",

3));

- How are the results of the program running, are there any changes ?

-The output is no change,it still same as before

-But in the code work it change ,it make new instance of Processor which is created with the same parameters ("Intel i5", 3)

- **Experiment 2**

Class car:

```java
public class Car {
    private String brand;
    private int cost;

    public Car (){

    }

    public Car(String brand,int cost){
        this.brand = brand;
        this.cost = cost;
    }

    public String getBrand(){
        return brand;
    }

    public void setBrand(String brand){
        this.brand = brand;
    }

    public int getCost(){
        return cost;
```

Class Customer:

```java
public class Customer {
    private String name;
    private Car car;
    private Driver driver;
    private int day;

    public Customer() {

    }


    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setCar(Car car) {
        this.car = car;
    }

    public Car getCar() {
        return car;
    }

    public void setDriver(Driver driver) {
        this.driver = driver;
    }

    public Driver getDriver() {
```

Class driver:

Driver.java > Driver > driverCostCalculation(int)

```java
public class Driver {
    private String name;
    private int cost;

    public Driver() {

    }

    public Driver(String name,int cost){
        this.name = name;
        this.cost = cost;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setCost(int cost) {
        this.cost = cost;
    }

    public int getCost() {
        return cost;
    }
```

Class main:

```java
public class MainExperiment2 {
    Run | Debug
    public static void main(String[] args) {

        Car c = new Car();
        c.setBrand(brand:"Avanza");
        c.setCost(cost:350000);


        Driver d = new Driver();
        d.setName(name:"John Doe");
        d.setCost(cost:200000);


        Customer cust = new Customer();
        cust.setName(name:"Jane Doe");
        cust.setCar(c);
        cust.setDriver(d);
        cust.setDay(day:2);

        System.out.println(x:"List payment travel: ");
        System.out.println("Customer Name: " + cust.getName());
        System.out.println("Price car " +c.getCost());
        System.out.println("Price for car for " + cust.getDay() + " days: " + c.carCostCalc

        System.out.println("Price for driver "+d.driverCostCalculation(days:2));
        System.out.println(x:"-------------------------");
        System.out.println("Total cost = " + cust.totalCostCalculation());
    }
}
```

Result:

```
List payment travel:
Customer Name: Jane Doe
Price car 350000
Price for car for 2 days: 700000
Price for driver 400000
-------------------------
Total cost = 1100000
PS D:\Semester 3\JAVA OOP\Week 5>
```

Questions

1. See the Customer Class. In Which program line that shows the Customer class has a relation with the Car class and Driver class ?

- private Car car;

  private Driver driver;

2. Pay attention to the method of calculating the Cost of Driver in the Driver class, and the method of calculating the Cost of a Car in the Car class. Why do you think that method must have a day argument ?

- because the total cost is dependent on the number of days the service

-To calculate the total cost, you need to multiply the cost per day by the number of days

3. Pay attention to the code from the Customer class. What do method for

car.carCostCalculation(day) and

driver.driverCostCalculation(day) ?

- for calculating the total cost of the car rental and driver hire, respectively, based on the number of days the customer

4. See the MainExperiment2 class. What are the code cust.setCar(c) and

cust.setDriver(d) ?

- cust.setCar(c): This set Car object with c (a car with the brand "Avanza") with the Customer object cust. It links the customer to the car they are renting.

- cust.setDriver(d): This set the Driver object with d (a driver named "John Doe") with the Customer object cust. It links the customer to the driver they are hiring.

5. See the MainExperiment2 class. What do method for

cust.totalCostCalculation() ?

- responsible for calculating the total cost that the customer incurs for renting a car and hiring a driver over a specified number of days.

6. See the MainExperiment2 class, try adding to the last line of the main method and observe the changes as they run!

System.out.println(cust.getCar().getBrand());

So what is the syntax for cust.getCar().getBrand() in the main method ?

```
List payment travel:
Customer Name: Jane Doe
Brand: Avanza
Price car 350000
Price for car for 2 days: 700000
Price for driver 400000
-------------------------
Total cost = 1100000
PS D:\Semester 3\JAVA OOP\Week 5>
```

-To get what brand car that customer order

- **Experiment 3**

Class Employee:

Employee.java > Employee > Employee(String, String)

```java
public class Employee {
    private String identityNumber;
    private String name;

    public Employee(String identityNumber,String name){
        this.identityNumber = identityNumber;
        this.name = name;
    }

    public String getIdentityNumber(){
        return this.identityNumber;
    }

    public void setIdentityNumber(String identityNumber){
        this.identityNumber = identityNumber;
    }

    public String getName(){
        return this.name;
    }

    public void setName(String name){
        this.name = name;
    }

    public String info() {
        String info = "";
        info += "Identity Number: " + this.identityNumber + "\n";
        info += "Name : " + this.name+ "\n";
        return info;
}
}
```

Class Railway:

```java
Railway.java > Railway > info()
public class Railway {
private String name;
private String railwayClass;
private Employee machinist;
private Employee assistant;

    public Railway(String name,String railwayClass,Employee machinist){
        this.name = name;
        this.railwayClass = railwayClass;
        this.machinist = machinist;
    }

    public Railway (String name,String railwayClass, Employee machinist,Employee assistant){
        this.name = name;
        this.railwayClass = railwayClass;
        this.machinist = machinist;
        this.assistant = assistant;
}

public String getName() {
    return name;
}


public void setName(String name) {
    this.name = name;
}
```

Class main:

```java
MainExperiment3.java > MainExperiment3 > main(String[])
public class MainExperiment3 {
    Run | Debug
    public static void main(String[] args) {
        Employee machinist = new Employee(identityNumber:"1234", name:"Spongebob Squarepants");
        Employee assistant = new Employee(identityNumber:"4567", name:"Patrick Star");
        Railway railway = new Railway(name:"Gaya Baru", railwayClass:"Bisnis", machinist, assistant);
        System.out.println(railway.info());

    }
}
```

Output:

```
Name: Gaya Baru
Railway Class: Bisnis
Machinist: Identity Number: 1234
Name : Spongebob Squarepants

Assistant: Identity Number: 4567
Name : Patrick Star


PS D:\Semester 3\JAVA OOP\Week 5>
```

Questions

1. The info() method in the Railway class, the line of code this.machinist.info() and this.assistant.info() is used for what ?

- to delegate the responsibility of printing or displaying specific details of the machinist and assistant employees to their respective Employee objects

2. Create a new main program with the name MainQuestion class in the same package. Add the following code to the main() method !

Employee machinist = new Employee("1234", "Spongebob Squarepants");

Railway railway = new Railway("Gaya Baru", "Bisnis", machinist);

System.out.println(railway.info());

-

```
MainQuestion3.java > ...
1   public class MainQuestion3 {
        Run | Debug
2       public static void main(String[] args) {
3
4           Employee machinist = new Employee(identityNumber:"1234", name:"Spongebob squarepants");
5           Railway railway = new Railway(name:"gaya baru", railwayClass:"Bisnis", machinist);
6           System.out.println(railway.info());
7       }
8   }
9
```

3. What is the output from the main program ? Why did this happen ?

-

```
Exception in thread "main" java.lang.NullPointerExcep
this.assistant" is null
        at Railway.info(Railway.java:64)
        at MainQuestion3.main(MainQuestion3.java:6)
PS D:\Semester 3\JAVA OOP\Week 5>
```

-because the railway info is include assistant info and assistant is null,that's why error

4. Fix the Railway class so the program can run !

```
public void setAssistant(Employee assistant) {
        this.assistant = assistant;
}

        public String info() {
            String info = "";
            info += "Name: " + this.name + "\n";
            info += "Railway Class: " + this.railwayClass + "\n";
            info += "Machinist: " + (this.machinist != null ? this.machinist.info() : "Machinist not assigned") + "\n";

            if (this.assistant != null) {
                info += "Assistant: " + this.assistant.info() + "\n";
            } else {
                info += "Assistant: Assistant not assigned\n";
            }

            return info;
        }


}
```

-in my opinion employee divided by 2,there is machinist and assistant,but because in this line :

```
public Railway(String name,String railwayClass,Employee machinist){
```

So I suggest,assistant is not must exist in that railway(train)

Output:

```
ta\Roaming\Code\User\workspaceStorag
Name: gaya baru
Railway Class: Bisnis
Machinist: Identity Number: 1234
Name : Spongebob squarepants

Assistant: Assistant not assigned

PS D:\Semester 3\JAVA OOP\Week 5>
```

- **Experiment 4**

Class Passenger:

Seat.java > Seat

```java
public class Seat {
    private String seatNumber;
    private Passenger passenger;

    public Seat(String seatNumber) {
        this.seatNumber = seatNumber;
    }

    public void setSeatNumber(String seatNumber) {
        this.seatNumber = seatNumber;
    }

    public String getSeatNumber() {
        return seatNumber;
    }

    public void setPassenger(Passenger passenger) {
        this.passenger = passenger;
    }

    public Passenger getPassenger() {
        return passenger;
    }

    public String info() {
        String info = "";
        info += String.format(format:"Seat Number: %s %n", seatNumber);
        if (this.passenger != null) {
            info += String.format(format:"Passenger: %s %n", passenger.info());
        }
        return info;
    }

```

Class carriage:

Carriage.java > Carriage > setPassenger(Passenger, int)

```java
public class Carriage {
    private String code;
    private Seat[] seats;

    private void initSeat() {
        for (int i = 0; i < seats.length; i++) {
            this.seats[i] = new Seat(String.valueOf(i + 1));
        }
    }

    public Carriage(String code, int amount) {
        this.code = code;
        this.seats = new Seat[amount];
        this.initSeat();
    }

    public void setCode(String code) {
        this.code = code;
    }

    public String getCode() {
        return code;
    }

    public void setPassenger(Passenger passenger, int number) {

        this.seats[number-1].setPassenger(passenger);


    }

    public Seat[] getSeats() {
        return seats;
```

Main:

```java
public class MainExperiment4 {
    Run | Debug
    public static void main(String[] args) {
        Passenger p = new Passenger(IDCard:"12345", name:"Mr. Krab");
        Carriage carriage = new Carriage(code:"A", amount:10);
        carriage.setPassenger(p, number:1);
        System.out.println(carriage.info());
    }
}
```

Output:

```
Code: A
Seat Number: 1
Passenger: ID Card: 12345
Name: Mr. Krab

Seat Number: 2
Seat Number: 3
Seat Number: 4
Seat Number: 5
Seat Number: 6
Seat Number: 7
Seat Number: 8
Seat Number: 9
Seat Number: 10

PS D:\Semester 3\JAVA OOP\Week 5>
```

Questions

1. In the MainExperiment4 class, what is amount of seats in carriage A ?

- 10 seats in the carriage A

2. Pay attention to the code snippet in the info() method in the Seat class. What does the

code mean ?

...

if (this.passenger != null) {

info += "Passenger: " + passenger.info() + "\n";

}

...

- The line if (this.passenger != null) checks if the passenger attribute of the Seat object is not null.if not null calls the info() method of the Passenger object: passenger.info()

-If the passenger is null, it means no passenger has been assigned to the seat. If it is not null, then a Passenger object is present in the seat.

3. Why the setPassengers() method in Carriage class, the value of number is reduced by the number 1 ?

-because first element in an array is at index 0, the second element is at index 1, and so on. In seat numbering in a train, the first seat would be labeled as seat 1, not seat 0.

By subtracting 1 from the number, the program maps the user-friendly seat number (starting from 1) to the correct zero-based index in the seats[] array.

4. Instantiation of new budi object with the Passenger type, then insert the new object in

the carriage with the carriage.setPassenger(budi, 1) method. What's happening ?

-error

```
carriage.setPassenger(passenger:"Budi", number:1);
```

5. Modify the program so that it is not allowed to occupy the seat of another passenger !

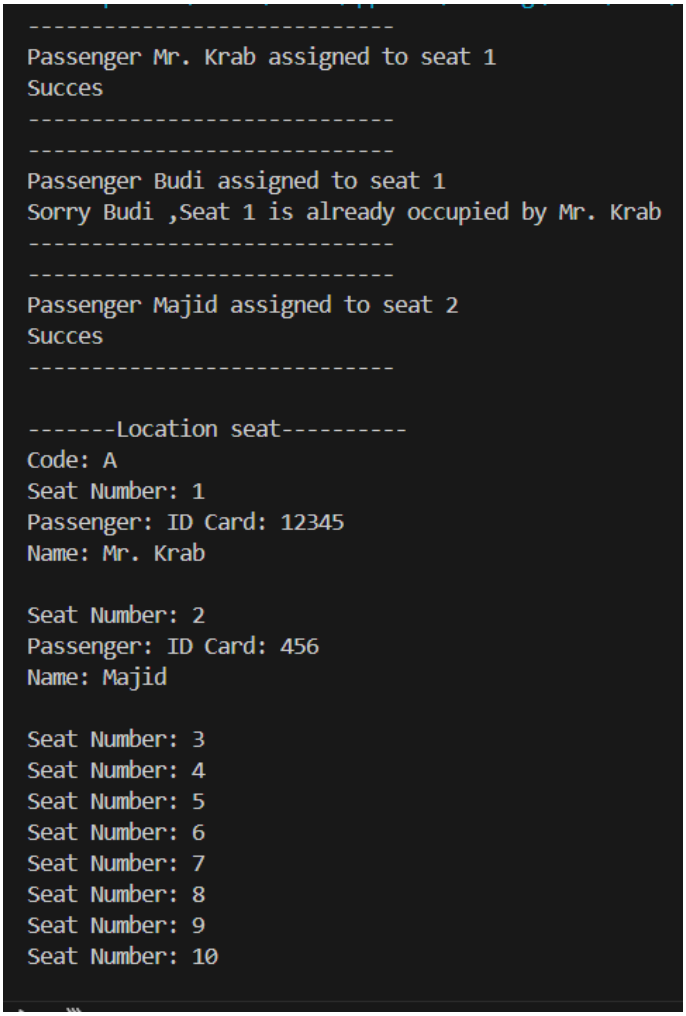
```java
public void setPassenger(Passenger passenger, int number) {
    if (seats[number - 1].getPassenger() == null) {
        this.seats[number - 1].setPassenger(passenger);
        System.out.println("Passenger " + passenger.getName() + " assigned to seat " + number);
        System.out.println(x:"Succes");
        System.out.println(x:"----------------------------");
    } else {
        System.out.println(x:"----------------------------");
        System.out.println("Passenger " + passenger.getName() + " assigned to seat " + number);
        System.out.println("Sorry " +passenger.getName() +" ,Seat " + number + " is already occupied by " + seats[number - 1].getPassenger().getNam
    }
}
```

Main:

MainExperiment4.java > ...

```java
public class MainExperiment4 {
    Run | Debug
    public static void main(String[] args) {
        Passenger p1 = new Passenger(IDCard:"12345", name:"Mr. Krab");
        Passenger p2 = new Passenger(IDCard:"125", name:"Budi");
        Passenger p3 = new Passenger(IDCard:"456", name:"Majid");
        Carriage carriage = new Carriage(code:"A", amount:10);
        carriage.setPassenger(p1, number:1);
        carriage.setPassenger(p2, number:1);
        carriage.setPassenger(p3, number:2);
        System.out.println(carriage.info());


    }
}
```

Output:

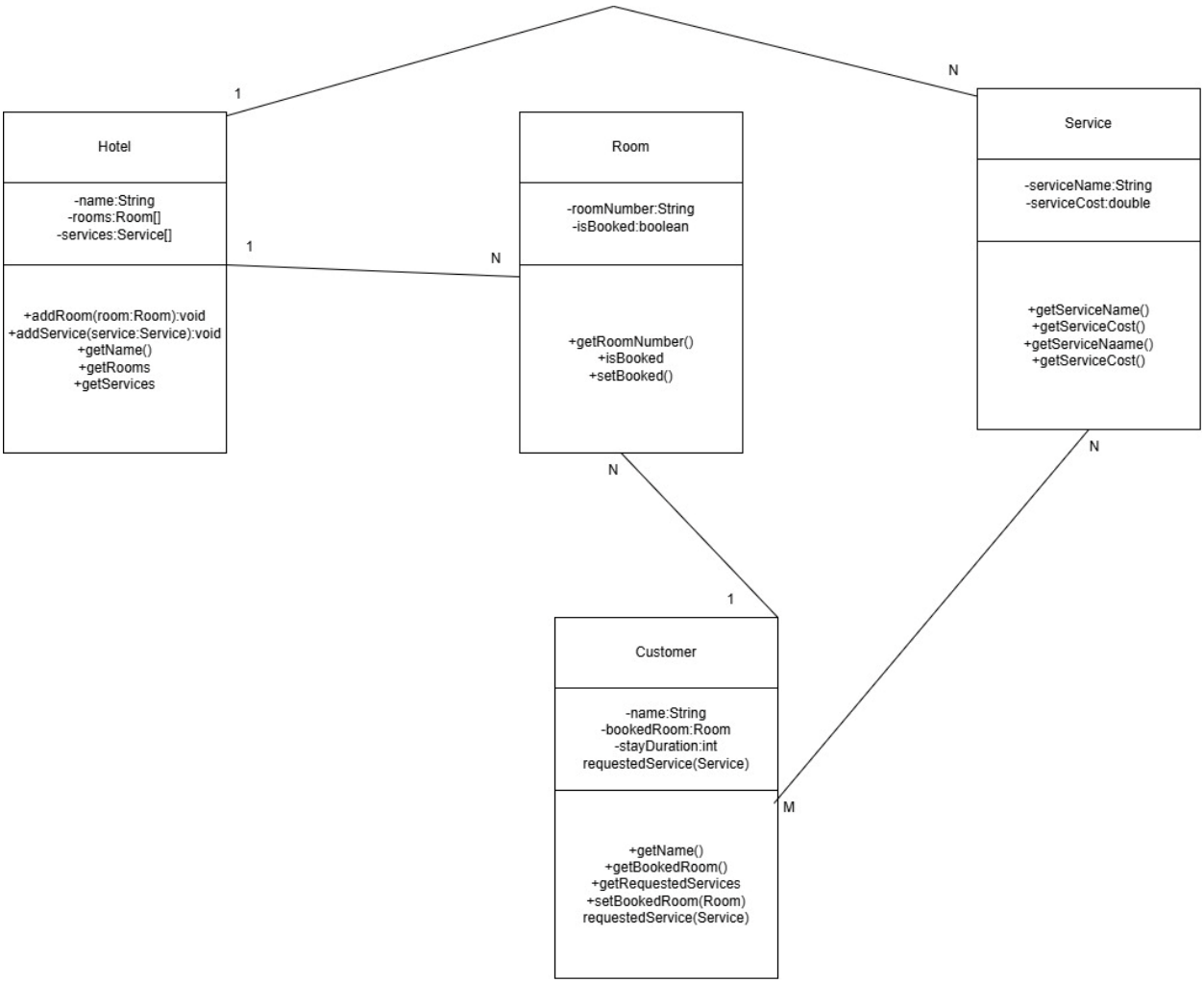


- **IV. Assignment**

Create a case study, design a class diagram, then implement in the code! The case studies should represent the relation class of experiments have been done on this matter, involving at least 4 classes (the main class does not count)

-System booking hotel sahid Montana

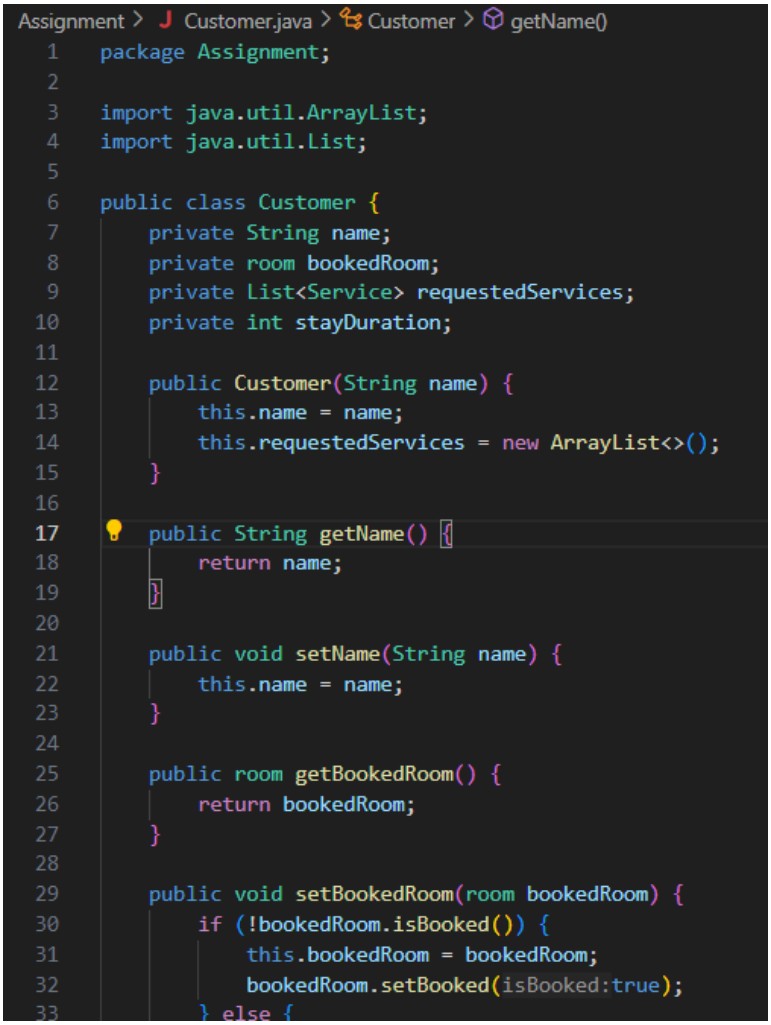
(Hotel,Room,Service,Customer)



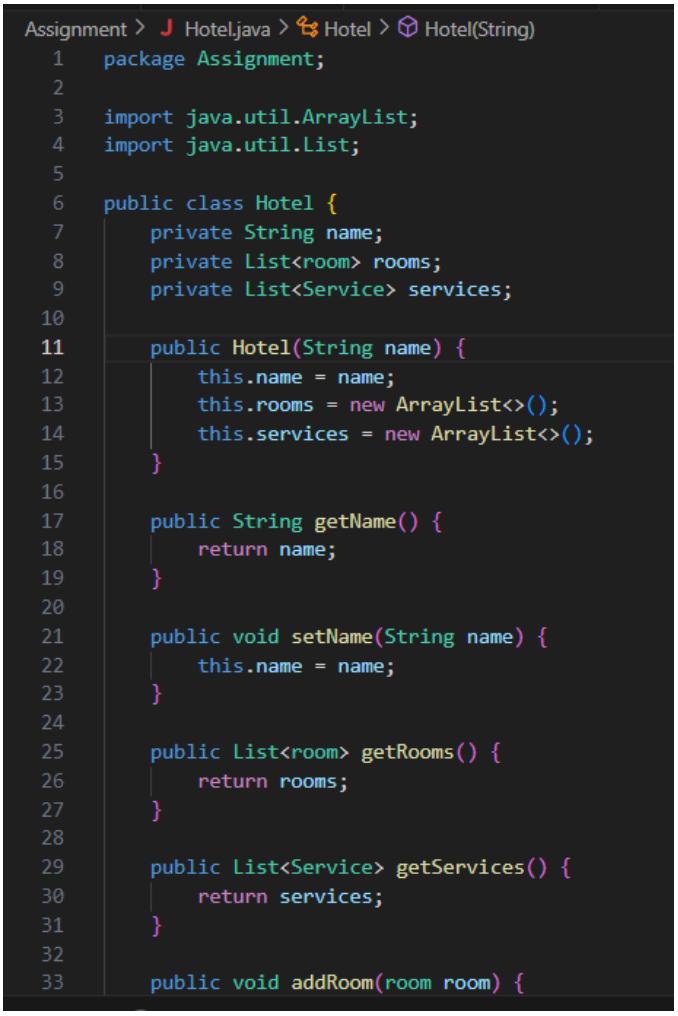
Output:

```
Hotel: Sahid Montana
Rooms:
Room Number: 1, Is Booked: No
Room Number: 2, Is Booked: No
Room Number: 3, Is Booked: No
Services:
Service: Double bed
Service: Swimming pool
Service: Breakfast

Customer: Majid
Booked Room: 3
Requested Services:
Service: Breakfast
Total Stay Duration: 2 days

Room Cost: 2 * 100_000 = 200000.0
Service Cost: Breakfast = 100000.0
Total Cost: 200000.0 + 100000.0 = 300000.0

Hotel: Sahid Montana
Rooms:
Room Number: 1, Is Booked: No
Room Number: 2, Is Booked: No
Room Number: 3, Is Booked: Yes
Services:
Service: Double bed
Service: Swimming pool
Service: Breakfast
```

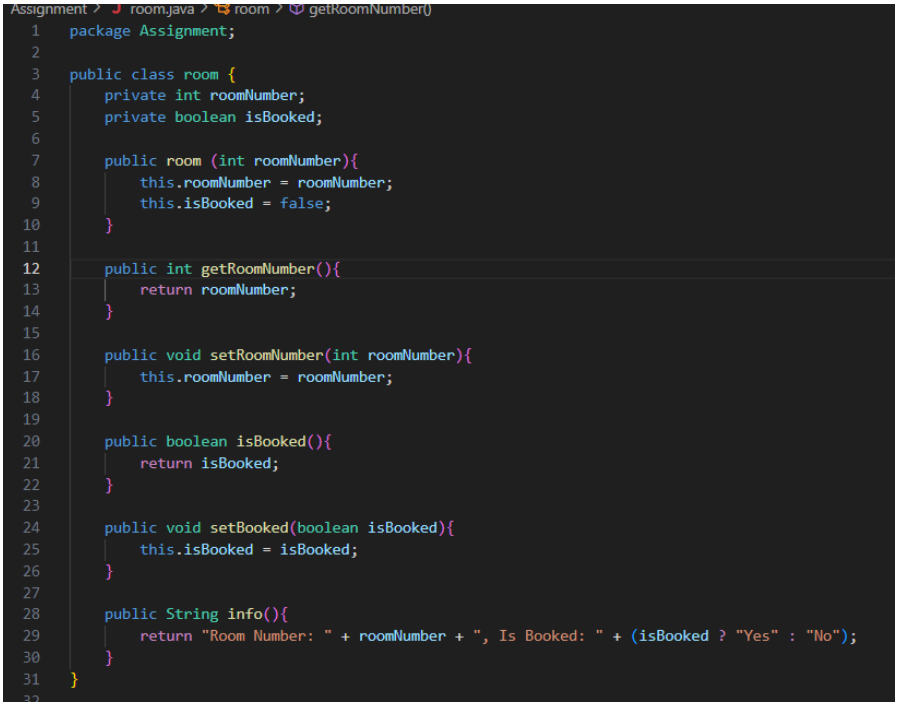
Customer:


Assignment > Customer.java > Customer > getName()
package Assignment;
import java.util.ArrayList;
import java.util.List;
public class Customer {
private String name;
private room bookedRoom;
private List<Service> requestedServices;
private int stayDuration;
public Customer(String name) {
this.name = name;
this.requestedServices = new ArrayList<>();
}
public String getName() {
return name;
}
public void setName(String name) {
this.name = name;
}
public room getBookedRoom() {
return bookedRoom;
}
public void setBookedRoom(room bookedRoom) {
if (!bookedRoom.isBooked()) {
this.bookedRoom = bookedRoom;
bookedRoom.setBooked(isBooked:true);
} else {

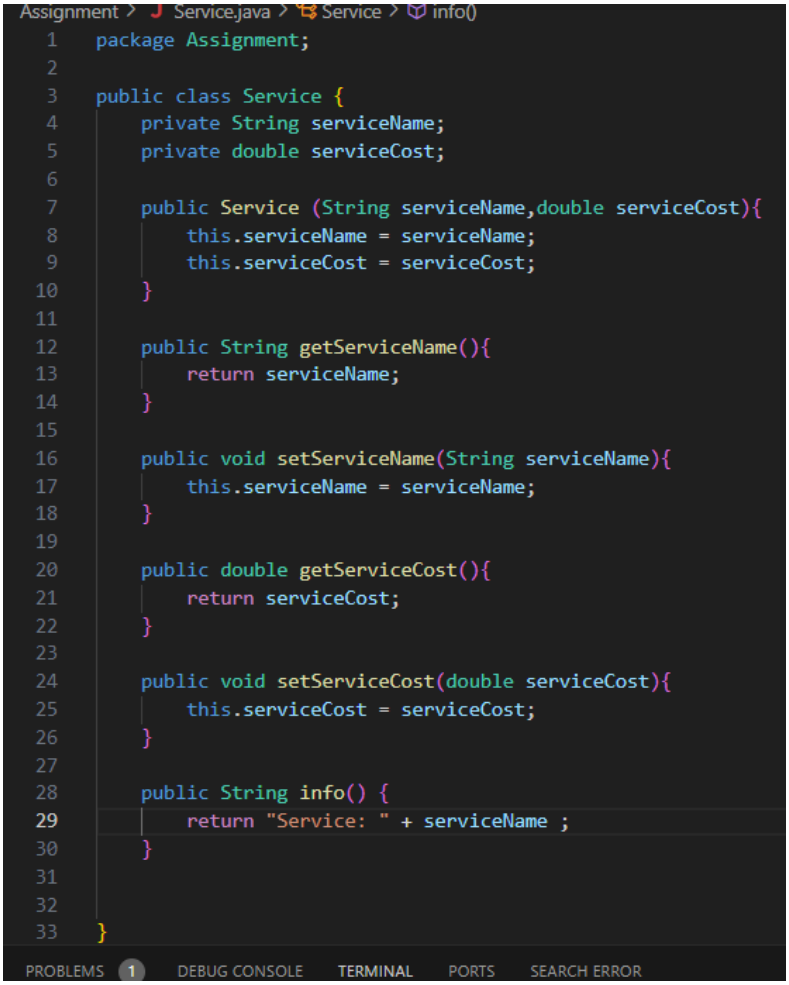
Hotel:


Assignment > Hotel.java > Hotel > Hotel(String)
package Assignment;
import java.util.ArrayList;
import java.util.List;
public class Hotel {
private String name;
private List<room> rooms;
private List<Service> services;
public Hotel(String name) {
this.name = name;
this.rooms = new ArrayList<>();
this.services = new ArrayList<>();
}
public String getName() {
return name;
}
public void setName(String name) {
this.name = name;
}
public List<room> getRooms() {
return rooms;
}
public List<Service> getServices() {
return services;
}
public void addRoom(room room) {

Room:


Assignment > room.java > room > getRoomNumber()
package Assignment;
public class room {
private int roomNumber;
private boolean isBooked;
public room (int roomNumber){
this.roomNumber = roomNumber;
this.isBooked = false;
}
public int getRoomNumber(){
return roomNumber;
}
public void setRoomNumber(int roomNumber){
this.roomNumber = roomNumber;
}
public boolean isBooked(){
return isBooked;
}
public void setBooked(boolean isBooked){
this.isBooked = isBooked;
}
public String info(){
return "Room Number: " + roomNumber + ", Is Booked: " + (isBooked ? "Yes" : "No");
}
}

Service:


Assignment > Service.java > Service > info()
1 package Assignment;
2
3 public class Service {
4 private String serviceName;
5 private double serviceCost;
6
7 public Service (String serviceName,double serviceCost){
8 this.serviceName = serviceName;
9 this.serviceCost = serviceCost;
10 }
11
12 public String getServiceName(){
13 return serviceName;
14 }
15
16 public void setServiceName(String serviceName){
17 this.serviceName = serviceName;
18 }
19
20 public double getServiceCost(){
21 return serviceCost;
22 }
23
24 public void setServiceCost(double serviceCost){
25 this.serviceCost = serviceCost;
26 }
27
28 public String info() {
29 return "Service: " + serviceName ;
30 }
31
32
33 }
PROBLEMS 1 DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

Main:

```java
package Assignment;
public class MainHotel {
    Run | Debug
    public static void main(String[] args) {

        Hotel sahidmontana = new Hotel(name:"Sahid Montana");
        Service breakfast = new Service(serviceName:"Breakfast", serviceCost:100_000);
        Service swimmingpool = new Service(serviceName:"Swimming pool", serviceCost:30_000);
        Service doublebed = new Service(serviceName:"Double bed", serviceCost:100_000);

        room standard = new room(roomNumber:01);
        room deluxe = new room(roomNumber:02);
        room suite = new room(roomNumber:03);

        sahidmontana.addRoom(standard);
        sahidmontana.addRoom(deluxe);
        sahidmontana.addRoom(suite);

        sahidmontana.addService(doublebed);
        sahidmontana.addService(swimmingpool);
        sahidmontana.addService(breakfast);



        System.out.println(sahidmontana.info());

        Customer cust1 = new Customer(name:"Majid");
        cust1.setBookedRoom(suite);
        cust1.requestService(breakfast);
        cust1.setStayDuration(stayDuration:2);
        System.out.println(cust1.info());
        cust1.printDetailedCost();


        System.out.println(sahidmontana.info());



    }
}

```

-full code on github