```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.Scanner;


class Food implements Serializable
{
int itemno;
int quantity;
float price;

Food(int itemno,int quantity)
{
this.itemno=itemno;
this.quantity=quantity;
switch(itemno)
{
case 1:price=quantity*50;
break;
case 2:price=quantity*60;
break;
case 3:price=quantity*70;
break;
case 4:price=quantity*30;
break;
}
}
}
```

```
class Singleroom implements Serializable
{
String name;
String contact;
String gender;
ArrayList<Food> food =new ArrayList<>();



Singleroom()
{
this.name="";
}
Singleroom(String name,String contact,String gender)
{
this.name=name;
this.contact=contact;
this.gender=gender;
}
}
class Doubleroom extends Singleroom implements Serializable
{
String name2;
String contact2;
String gender2;

Doubleroom()
{
this.name="";
this.name2="";
}
Doubleroom(String name,String contact,String gender,String name2,String contact2,String gender2)
{
```

```
this.name=name;
this.contact=contact;
this.gender=gender;
this.name2=name2;
this.contact2=contact2;
this.gender2=gender2;
}
}
class NotAvailable extends Exception
{
@Override
public String toString()
{
return "Not Available !";
}
}


class holder implements Serializable
{
Doubleroom luxury_doublerrom[]=new Doubleroom[10]; //Luxury
Doubleroom deluxe_doublerrom[]=new Doubleroom[20]; //Deluxe
Singleroom luxury_singleerrom[]=new Singleroom[10]; //Luxury
Singleroom deluxe_singleerrom[]=new Singleroom[20]; //Deluxe
}


class Hotel
{
static holder hotel_ob=new holder();
static Scanner sc = new Scanner(System.in);
static void CustDetails(int i,int rn)
{
```

```
String name, contact, gender;
String name2 = null, contact2 = null;
String gender2="";
System.out.print("\nEnter customer name: ");
name = sc.next();
System.out.print("Enter contact number: ");
contact=sc.next();
System.out.print("Enter gender: ");
gender = sc.next();
if(i<3)
{
System.out.print("Enter second customer name: ");
name2 = sc.next();
System.out.print("Enter contact number: ");
contact2=sc.next();
System.out.print("Enter gender: ");
gender2 = sc.next();
}

switch (i) {
case 1:hotel_ob.luxury_doublerrom[rn]=new Doubleroom(name,contact,gender,name2,contact2,gender2);
break;
case 2:hotel_ob.deluxe_doublerrom[rn]=new Doubleroom(name,contact,gender,name2,contact2,gender2);
break;
case 3:hotel_ob.luxury_singleerrom[rn]=new Singleroom(name,contact,gender);
break;
case 4:hotel_ob.deluxe_singleerrom[rn]=new Singleroom(name,contact,gender);
break;
default:System.out.println("Wrong option");
break;
}
}
```

```
static void bookroom(int i)
{
int j;
int rn;
System.out.println("\nChoose room number from : ");
switch (i) {
case 1:
for(j=0;j<hotel_ob.luxury_doublerrom.length;j++)
{
if(hotel_ob.luxury_doublerrom[j]==null)
{
System.out.print(j+1+",");
}
}
System.out.print("\nEnter room number: ");
try{
rn=sc.nextInt();
rn--;
if(hotel_ob.luxury_doublerrom[rn]!=null)
throw new NotAvailable();
CustDetails(i,rn);
}
catch(Exception e)
{
System.out.println("Invalid Option");
return;
}
break;
case 2:
for(j=0;j<hotel_ob.deluxe_doublerrom.length;j++)
{
if(hotel_ob.deluxe_doublerrom[j]==null)
{
```

```
System.out.print(j+11+",");
}
}
System.out.print("\nEnter room number: ");
try{
rn=sc.nextInt();
rn=rn-11;
if(hotel_ob.deluxe_doublerrom[rn]!=null)
throw new NotAvailable();
CustDetails(i,rn);
}
catch(Exception e)
{
System.out.println("Invalid Option");
return;
}
break;
case 3:
for(j=0;j<hotel_ob.luxury_singleerrom.length;j++)
{
if(hotel_ob.luxury_singleerrom[j]==null)
{
System.out.print(j+31+",");
}
}
System.out.print("\nEnter room number: ");
try{
rn=sc.nextInt();
rn=rn-31;
if(hotel_ob.luxury_singleerrom[rn]!=null)
throw new NotAvailable();
CustDetails(i,rn);
}
```

```
catch(Exception e)
{
System.out.println("Invalid Option");
return;
}
break;
case 4:
for(j=0;j<hotel_ob.deluxe_singleerrom.length;j++)
{
if(hotel_ob.deluxe_singleerrom[j]==null)
{
System.out.print(j+41+",");
}
}
System.out.print("\nEnter room number: ");
try{
rn=sc.nextInt();
rn=rn-41;
if(hotel_ob.deluxe_singleerrom[rn]!=null)
throw new NotAvailable();
CustDetails(i,rn);
}
catch(Exception e)
{
System.out.println("Invalid Option");
return;
}
break;
default:
System.out.println("Enter valid option");
break;
}
System.out.println("Room Booked");
```

```
}

static void features(int i)
{
switch (i) {
case 1:System.out.println("Number of double beds : 1\nAC : Yes\nFree breakfast : Yes\nCharge per day:4000 ");
break;
case 2:System.out.println("Number of double beds : 1\nAC : No\nFree breakfast : Yes\nCharge per day:3000 ");
break;
case 3:System.out.println("Number of single beds : 1\nAC : Yes\nFree breakfast : Yes\nCharge per day:2200 ");
break;
case 4:System.out.println("Number of single beds : 1\nAC : No\nFree breakfast : Yes\nCharge per day:1200 ");
break;
default:
System.out.println("Enter valid option");
break;
}
}

static void availability(int i)
{
int j,count=0;
switch (i) {
case 1:
for(j=0;j<10;j++)
{
if(hotel_ob.luxury_doublerrom[j]==null)
count++;
}
break;
case 2:
for(j=0;j<hotel_ob.deluxe_doublerrom.length;j++)
{
```

```
if(hotel_ob.deluxe_doublerrom[j]==null)
count++;
}
break;
case 3:
for(j=0;j<hotel_ob.luxury_singleerrom.length;j++)
{
if(hotel_ob.luxury_singleerrom[j]==null)
count++;
}
break;
case 4:
for(j=0;j<hotel_ob.deluxe_singleerrom.length;j++)
{
if(hotel_ob.deluxe_singleerrom[j]==null)
count++;
}
break;
default:
System.out.println("Enter valid option");
break;
}
System.out.println("Number of rooms available : "+count);
}

static void bill(int rn,int rtype)
{
double amount=0;
String list[]={"Sandwich","Pasta","Noodles","Coke"};
System.out.println("\n*******");
System.out.println(" Bill:-");
System.out.println("*******");
```

```
switch(rtype)
{
case 1:
amount+=4000;
System.out.println("\nRoom Charge - "+4000);
System.out.println("\n================");
System.out.println("Food Charges:- ");
System.out.println("================");
System.out.println("Item Quantity Price");
System.out.println("-------------------------");
for(Food obb:hotel_ob.luxury_doublerrom[rn].food)
{
amount+=obb.price;
String format = "%-10s%-10s%-10s%n";
System.out.printf(format,list[obb.itemno-1],obb.quantity,obb.price );
}

break;
case 2:amount+=3000;
System.out.println("Room Charge - "+3000);
System.out.println("\nFood Charges:- ");
System.out.println("================");
System.out.println("Item Quantity Price");
System.out.println("-------------------------");
for(Food obb:hotel_ob.deluxe_doublerrom[rn].food)
{
amount+=obb.price;
String format = "%-10s%-10s%-10s%n";
System.out.printf(format,list[obb.itemno-1],obb.quantity,obb.price );
}
break;
case 3:amount+=2200;
System.out.println("Room Charge - "+2200);
System.out.println("\nFood Charges:- ");
System.out.println("===============");
System.out.println("Item Quantity Price");
System.out.println("—----------------------");
for(Food obb:hotel_ob.luxury_singleerrom[rn].food)
{
amount+=obb.price;
String format = "%-10s%-10s%-10s%n";
System.out.printf(format,list[obb.itemno-1],obb.quantity,obb.price );
}
break;
case 4:amount+=1200;
System.out.println("Room Charge - "+1200);
System.out.println("\nFood Charges:- ");
System.out.println("===============");
System.out.println("Item Quantity Price");
System.out.println("—----------------------");
for(Food obb: hotel_ob.deluxe_singleerrom[rn].food)
{
amount+=obb.price;
String format = "%-10s%-10s%-10s%n";
System.out.printf(format,list[obb.itemno-1],obb.quantity,obb.price );
}
break;
default:
System.out.println("Not valid");
}
System.out.println("\nTotal Amount- "+amount);
}

static void deallocate(int rn,int rtype)
{
int j;
```

```
char w;
switch (rtype) {
case 1:
if(hotel_ob.luxury_doublerrom[rn]!=null)
System.out.println("Room used by "+hotel_ob.luxury_doublerrom[rn].name);
else
{
System.out.println("Empty Already");
return;
}
System.out.println("Do you want to checkout ?(y/n)");
w=sc.next().charAt(0);
if(w=='y'||w=='Y')
{
bill(rn,rtype);
hotel_ob.luxury_doublerrom[rn]=null;
System.out.println("Deallocated succesfully");
}

break;
case 2:
if(hotel_ob.deluxe_doublerrom[rn]!=null)
System.out.println("Room used by "+hotel_ob.deluxe_doublerrom[rn].name);
else
{
System.out.println("Empty Already");
return;
}
System.out.println(" Do you want to checkout ?(y/n)");
w=sc.next().charAt(0);
if(w=='y'||w=='Y')
{
bill(rn,rtype);
hotel_ob.deluxe_doublerrom[rn]=null;
System.out.println("Deallocated succesfully");
}

break;
case 3:
if(hotel_ob.luxury_singleerrom[rn]!=null)
System.out.println("Room used by "+hotel_ob.luxury_singleerrom[rn].name);
else
{
System.out.println("Empty Already");
return;
}
System.out.println(" Do you want to checkout ? (y/n)");
w=sc.next().charAt(0);
if(w=='y'||w=='Y')
{
bill(rn,rtype);
hotel_ob.luxury_singleerrom[rn]=null;
System.out.println("Deallocated succesfully");
}

break;
case 4:
if(hotel_ob.deluxe_singleerrom[rn]!=null)
System.out.println("Room used by "+hotel_ob.deluxe_singleerrom[rn].name);
else
{
System.out.println("Empty Already");
return;
}
System.out.println(" Do you want to checkout ? (y/n)");
w=sc.next().charAt(0);
```

```java
if(w=='y'||w=='Y')
{
bill(rn,rtype);
hotel_ob.deluxe_singleerrom[rn]=null;
System.out.println("Deallocated succesfully");
}
break;
default:
System.out.println("\nEnter valid option : ");
break;
}
}

static void order(int rn,int rtype)
{
int i,q;
char wish;
try{
System.out.println("\n==========\n Menu: \n==========\n\n1.Sandwich\tRs.50\n2.Pasta\t\tRs.60\n3.Noodles\tRs.70\n4.Coke\t\tRs.30\n");
do
{
i = sc.nextInt();
System.out.print("Quantity- ");
q=sc.nextInt();

switch(rtype){
case 1: hotel_ob.luxury_doublerrom[rn].food.add(new Food(i,q));
break;
case 2: hotel_ob.deluxe_doublerrom[rn].food.add(new Food(i,q));
break;
case 3: hotel_ob.luxury_singleerrom[rn].food.add(new Food(i,q));
break;
case 4: hotel_ob.deluxe_singleerrom[rn].food.add(new Food(i,q));
break;
}
System.out.println("Do you want to order anything else ? (y/n)");
wish=sc.next().charAt(0);
}while(wish=='y'||wish=='Y');
}
catch(NullPointerException e)
{
System.out.println("\nRoom not booked");
}
catch(Exception e)
{
System.out.println("Cannot be done");
}
}
}



class write implements Runnable
{
holder hotel_ob;
write(holder hotel_ob)
{
this.hotel_ob=hotel_ob;
}
@Override
public void run() {
try{
FileOutputStream fout=new FileOutputStream("backup");
ObjectOutputStream oos=new ObjectOutputStream(fout);
oos.writeObject(hotel_ob);
}
```

```
catch(Exception e)
{
System.out.println("Error in writing "+e);
}

}

}


public class Main {
public static void main(String[] args){

try
{
File f = new File("backup");
if(f.exists())
{
FileInputStream fin=new FileInputStream(f);
ObjectInputStream ois=new ObjectInputStream(fin);
Hotel.hotel_ob=(holder)ois.readObject();
}
Scanner sc = new Scanner(System.in);
int ch,ch2;
char wish;
x:
do{


System.out.println("\nEnter your choice :\n1.Display room details\n2.Display room availability \n3.Book\n4.Order food\n5.Checkout\n6.Exit\n");
ch = sc.nextInt();
switch(ch){
case 1: System.out.println("\nChoose room type :\n1.Luxury Double Room \n2.Deluxe Double Room \n3.Luxury Single Room \n4.Deluxe Single
Room\n");
ch2 = sc.nextInt();
Hotel.features(ch2);
break;
case 2:System.out.println("\nChoose room type :\n1.Luxury Double Room \n2.Deluxe Double Room \n3.Luxury Single Room\n4.Deluxe Single Room\n");
ch2 = sc.nextInt();
Hotel.availability(ch2);
break;
case 3:System.out.println("\nChoose room type :\n1.Luxury Double Room \n2.Deluxe Double Room \n3.Luxury Single Room\n4.Deluxe Single Room\n");
ch2 = sc.nextInt();
Hotel.bookroom(ch2);
break;
case 4:
System.out.print("Room Number -");
ch2 = sc.nextInt();
if(ch2>60)
System.out.println("Room doesn't exist");
else if(ch2>40)
Hotel.order(ch2-41,4);
else if(ch2>30)
Hotel.order(ch2-31,3);
else if(ch2>10)
Hotel.order(ch2-11,2);
else if(ch2>0)
Hotel.order(ch2-1,1);
else
System.out.println("Room doesn't exist");
break;
case 5:
System.out.print("Room Number -");
ch2 = sc.nextInt();
```

```
if(ch2>60)
System.out.println("Room doesn't exist");
else if(ch2>40)
Hotel.deallocate(ch2-41,4);
else if(ch2>30)
Hotel.deallocate(ch2-31,3);
else if(ch2>10)
Hotel.deallocate(ch2-11,2);
else if(ch2>0)
Hotel.deallocate(ch2-1,1);
else
System.out.println("Room doesn't exist");
break;
case 6:break x;

}

System.out.println("\nContinue : (y/n)");
wish=sc.next().charAt(0);
if(!(wish=='y'||wish=='Y'||wish=='n'||wish=='N'))
{
System.out.println("Invalid Option");
System.out.println("\nContinue : (y/n)");
wish=sc.next().charAt(0);
}

}while(wish=='y'||wish=='Y');

Thread t=new Thread(new write(Hotel.hotel_ob));
t.start();
}
catch(Exception e)
{
```

```
System.out.println("Not a valid input");
}
}
}
```