**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.Date;

**import** java.util.List;

**import** java.util.Scanner;

**public** **class** HotelCasereq5{

**public** **static** **void** main(String[] args) **throws** NumberFormatException, ParseException {

List<Room>roomList=**new** ArrayList<Room>();

SimpleDateFormat df=**new** SimpleDateFormat("dd-MM-yyyy HH:mm:ss");

System.***out***.println("Enter the number of the rooms");

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

**int** no=Integer.*parseInt*(input.nextLine());

**for**(**int** i=0;i<no;i++)

{

String detail=input.nextLine();

Room room=Room.*CreateRoom*(detail);

roomList.add(room);

}

System.***out***.println("Enter a type of Sort \n1.Sort by Capacity\n2.Sort by Price\n3.Sort by Booked Time");

**int** choice=Integer.*parseInt*(input.nextLine());

**while**(choice>0 && choice<4)

{

**if**(choice==1)

{

Collections.*sort*(roomList);

System.***out***.println("Number\tFloor\tType\tCapacity\tBooked Time\tPrice");

**for** (Room room : roomList) {

System.***out***.println(room.get\_number()+"\t"+room.get\_floor()+"\t"+room.get\_type()+"\t"+room.get\_capacity()+"\t"+

df.format(room.get\_bookedTime())+"\t"+room.get\_price());

}

}

**if**(choice==2)

{

BookedTimeComparer btc=**new** BookedTimeComparer();

Collections.*sort*(roomList,btc);

System.***out***.println("Number\tFloor\tType\tCapacity\tBooked Time\tPrice");

**for** (Room room : roomList) {

System.***out***.println(room.get\_number()+"\t"+room.get\_floor()+"\t"+room.get\_type()+"\t"+room.get\_capacity()+"\t"+

df.format(room.get\_bookedTime())+"\t"+room.get\_price());

}

}

**if**(choice==3)

{

PriceComparer pc=**new** PriceComparer();

Collections.*sort*(roomList,pc);

System.***out***.println("Number\tFloor\tType\tCapacity\tBooked Time\tPrice");

**for** (Room room : roomList) {

System.***out***.println(room.get\_number()+"\t"+room.get\_floor()+"\t"+room.get\_type()+"\t"+room.get\_capacity()+"\t"+

df.format(room.get\_bookedTime())+"\t"+room.get\_price());

}

}

System.***out***.println("Enter a type of Sort \n1.Sort by Capacity\n2.Sort by Price\n3.Sort by Booked Time");

choice=Integer.*parseInt*(input.nextLine());

}

}

}

**class** Room **implements** Comparable<Room>

{

**private** **int** \_number;

**private** **int** \_floor;

**private** String \_type;

**private** **int** \_capacity;

**private** Date \_bookedTime;

**private** **double** \_price;

**public** **int** get\_number() {

**return** \_number;

}

**public** **void** set\_number(**int** \_number) {

**this**.\_number = \_number;

}

**public** **int** get\_floor() {

**return** \_floor;

}

**public** **void** set\_floor(**int** \_floor) {

**this**.\_floor = \_floor;

}

**public** String get\_type() {

**return** \_type;

}

**public** **void** set\_type(String \_type) {

**this**.\_type = \_type;

}

**public** **int** get\_capacity() {

**return** \_capacity;

}

**public** **void** set\_capacity(**int** \_capacity) {

**this**.\_capacity = \_capacity;

}

**public** Date get\_bookedTime() {

**return** \_bookedTime;

}

**public** **void** set\_bookedTime(Date \_bookedTime) {

**this**.\_bookedTime = \_bookedTime;

}

**public** **double** get\_price() {

**return** \_price;

}

**public** **void** set\_price(**double** \_price) {

**this**.\_price = \_price;

}

**public** Room(**int** \_number, **int** \_floor, String \_type, **int** \_capacity, Date \_bookedTime, **double** \_price) {

**this**.\_number = \_number;

**this**.\_floor = \_floor;

**this**.\_type = \_type;

**this**.\_capacity = \_capacity;

**this**.\_bookedTime = \_bookedTime;

**this**.\_price = \_price;

}

**public** Room() {

}

@Override

**public** String toString() {

SimpleDateFormat df=**new** SimpleDateFormat("dd-MM-yyyy HH:mm:ss");

**return** "Number:" + \_number +"\n"+ "Floor:" + \_floor +"\n"+ "Type:" + \_type +"\n"+ "Capacity:" + \_capacity+"\n"

+"BookedTime:" + df.format(\_bookedTime) +"\n"+ "Price:" + \_price+"\n";

}

**static** Room CreateRoom(String detail) **throws** NumberFormatException, ParseException

{

SimpleDateFormat df=**new** SimpleDateFormat("dd-MM-yyyy HH:mm:ss");

Room room=**new** Room(Integer.*parseInt*(detail.split(",")[0]),Integer.*parseInt*(detail.split(",")[1]),detail.split(",")[2],Integer.*parseInt*(detail.split(",")[3]),

df.parse(detail.split(",")[4]),Double.*parseDouble*(detail.split(",")[5]));

**return** room;

}

@Override

**public** **int** compareTo(Room r1) {

**if**(\_capacity>r1.\_capacity)

{

**return** 1;

}

**else** **if**(\_capacity <r1.\_capacity)

{

**return** -1;

}

**else**

{

**return** 0;

}

}

}

**class** BookedTimeComparer **implements** Comparator<Room>

{

@Override

**public** **int** compare(Room r1, Room r2) {

**return** r1.get\_bookedTime().compareTo(r2.get\_bookedTime());

}

}

**class** PriceComparer **implements** Comparator<Room>

{

@Override

**public** **int** compare(Room r1, Room r2) {

**if**(r1.get\_price() > r2.get\_price())

{

**return** 1;

}

**else** **if**(r1.get\_price() < r2.get\_price())

{

**return** -1;

}

**else**

{

**return** 0;

}

}

}