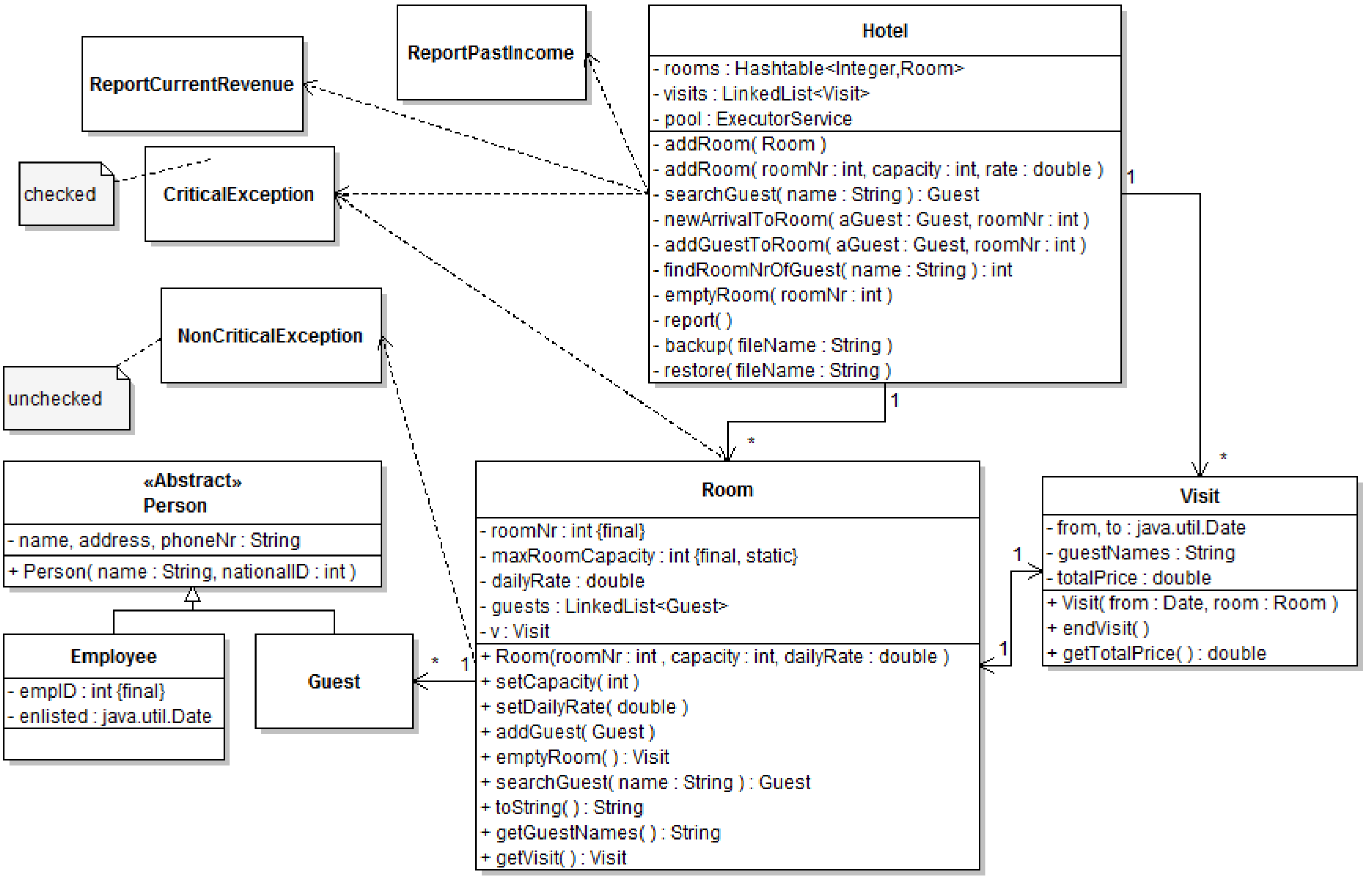
BLM2562 OBJECT ORIENTED PROG., Section 1&2, FINAL MAKEUP EXAM (06/07/2017)

Duration:	90 mins.			**Score:**					**Student Nr:**	**Signature:**
Grading:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Name, Surname:**	
	10	15	25	25	25					

QUESTIONS


ReportPastIncome
ReportCurrentRevenue
Hotel
- rooms : Hashtable<Integer,Room>
- visits : LinkedList<Visit>
- pool : ExecutorService
- addRoom(Room)
- addRoom(roomNr : int, capacity : int, rate : double)
- searchGuest(name : String) : Guest
- newArrivalToRoom(aGuest : Guest, roomNr : int)
- addGuestToRoom(aGuest : Guest, roomNr : int)
- findRoomNrOfGuest(name : String) : int
- emptyRoom(roomNr : int)
- report( )
- backup(fileName : String)
- restore(fileName : String)
checked
CriticalException
NonCriticalException
unchecked
«Abstract»
Person
- name, address, phoneNr : String
+ Person(name : String, nationalID : int)
Employee
- empID : int {final}
- enlisted : java.util.Date
Guest
Room
- roomNr : int {final}
- maxRoomCapacity : int {final, static}
- dailyRate : double
- guests : LinkedList<Guest>
- v : Visit
+ Room(roomNr : int , capacity : int, dailyRate : double)
+ setCapacity(int)
+ setDailyRate(double)
+ addGuest(Guest)
+ emptyRoom( ) : Visit
+ searchGuest(name : String) : Guest
+ toString( ) : String
+ getGuestNames( ) : String
+ getVisit( ) : Visit
Visit
- from, to : java.util.Date
- guestNames : String
- totalPrice : double
+ Visit(from : Date, room : Room)
+ endVisit( )
+ getTotalPrice( ) : double
1
*
1
*
1
*
1
1
1

Answer these questions according to the UML class schema given above. You may need to extract hidden information from the schema and add necessary code. You don’t have to consider inconsistencies other than the ones explicitly stated in questions. You will be writing some part of hotel management software.

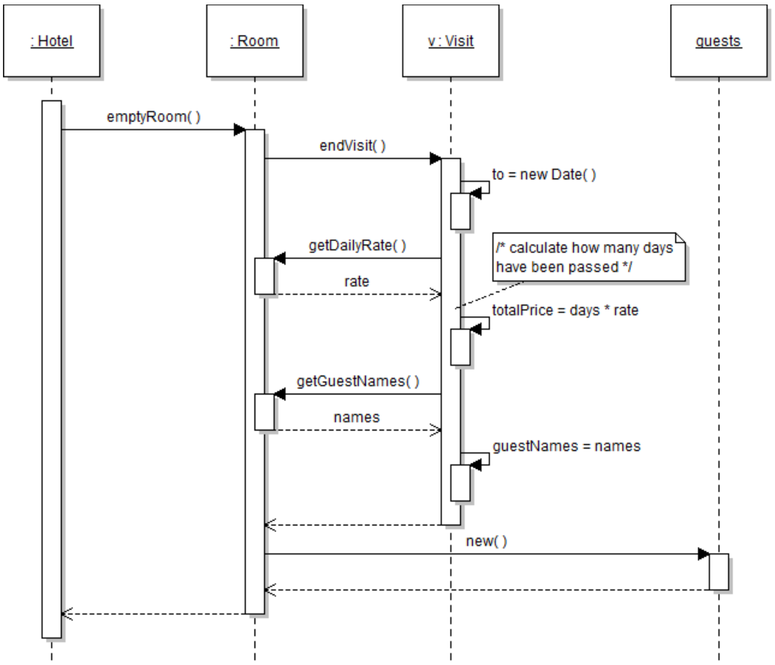
Pay attention to the following details about the program logic when writing your code:

Question 1: Write the source code of classes CriticalException and NonCriticalException.

Question 2: Write the source code of classes Person and Employee.

Question 3: Write the source code of class Room. The details of its methods are as follows:

- setCapacity: If one attempts to set a negative capacity or a capacity greater than maxRoomCapacity, a CriticalException is generated.
- setDailyRate: A room’s daily rate must not be changed while there are guests in it. Otherwise, a NonCriticalException is generated.
- Constructor must use the setter methods described above.
- addGuest: A guest must not be added multiple times. Otherwise, a CriticalException is generated.
- emptyRoom: Code it according to the given sequence diagram on the second paper.
- toString: Returns a String that contains information about the room number, its capacity and its current guests’ names.



Question 4: Write the source code of the classes ReportCurrentRevenue and ReportPastIncome. The ReportCurrentRevenue class is responsible from calculating the current amount of money to be gained if all the guests would leave from the hotel. The ReportPastIncome class is responsible from calculating the money gained so far from the previous visits of guests. Class Hotel will create one instance of this class and execute it in multithreaded fashion in its report method. You will code Hotel.report in the next question.

Question 5: Write the source code of only the following methods of the class Hotel. The explanations of the methods you need to code are given below. Don’t forget to address the multithreading issues when coding.

- report: The details of this method are given in Question 4.
- restore: This method loads the Hashtable members and LinkedList visits from the path given by the fileName parameter. However, if there are ongoing multithreaded operations, it must first wait them to be finished.
- findRoomNrOfGuest: This method returns the room number of a guest whose name is given by the method parameter. It returns 0 if no such guest exists.

BLM2562 OBJECT ORIENTED PROG., Section 1&2, FINAL MAKEUP EXAM (06/07/2017)

Question 1: Write the source code of classes CriticalException and NonCriticalException. (10p)

```
public class CriticalException extends java.io.IOException {
   public CriticalException(String arg0) {
      super(arg0);
   }
}
public class NonCriticalException extends RuntimeException {
   public NonCriticalException(String arg0) {
      super(arg0);
   }
}
```

Question 2: Write the source code of classes Person and Employee. (15p)

```
public abstract class Person implements java.io.Serializable {
   private String name, address, phoneNr;
   private int nationalID;

   public Person(String name, int nationalID) {
      this.name = name; this.nationalID = nationalID;
   }
   public void setAddress(String address) { this.address = address; }
   public void setPhoneNr(String phoneNr) { this.phoneNr = phoneNr; }
   public String getAddress() { return address; }
   public String getPhoneNr() { return phoneNr; }
   public String getName() { return name; }
   public int getNationalID() { return nationalID; }
   public String toString() {
      String str = "Name: " + name + ", National ID: " + nationalID;
      return str;
   }
}
public class Employee extends Person {
   private static final long serialVersionUID = 1L;
   private final int empID;
   private Date enlisted;

   public Employee(String name, int nationalID, int empID) {
      super(name, nationalID);
      this.empID = empID;
      enlisted = new Date();
   }
   public int getEmpID() { return empID; }
   public Date getEnlisted() { return enlisted; }
}
```

Question 3: Write the source code of class Room. The details of its methods are as follows: (25p)

```
package final_02;
import java.util.*;
public class Room implements java.io.Serializable{
   private static final long serialVersionUID = 1L;
   private final int roomNr;
   public final static int maxRoomCapacity = 10;
   private int capacity;
   private double dailyRate;
   private LinkedList<Guest> guests;
   private Visit v;
   public Room(int roomNr, int capacity, double dailyRate) throws CriticalException {
      this.roomNr = roomNr;
      setCapacity( capacity );  //or add try-catch
      setDailyRate( dailyRate );
      guests = new LinkedList<Guest>();
   }
   public int getRoomNr() { return roomNr; }
   public int getCapacity() { return capacity; }
   public int getGuestCount() { return guests.size(); }
   public double getDailyRate() { return dailyRate; }
   public void setCapacity( int newCapacity ) throws CriticalException {
      if( newCapacity <= 0 || newCapacity > maxRoomCapacity )
         throw new CriticalException("invalid capacity: " + newCapacity);
   }
   public void setDailyRate(double dailyRate) throws NonCriticalException {
      if( guests.size() == 0 )
         this.dailyRate = dailyRate;
      else throw new NonCriticalException
         ("Cannot change rate when room is already occupied");
   }
   public void addGuest(Guest guest) throws CriticalException {
      for( Guest existing : guests )
         if( existing == guest )
            throw new CriticalException("Duplicate guest: "+guest);
      if( guests.isEmpty() ) //or: if( v == null )
         v = new Visit(new Date(), this);
      guests.add(guest);
   }
   public Visit emptyRoom() {
      v.endVisit();
      guests = new LinkedList<Guest>();
      return v;
   }
   public Guest searchGuest(String name) {
      for( Guest existing : guests )
         if( existing.getName() == name )
            return existing;
      return null;
   }
   public String toString() {
      String str = "Room Nr: " + roomNr + "\nCapacity: " + capacity + "\nGuests: ";
      for( Guest existing : guests )
         str += "\n\t"+existing.toString();
      return str;
   }
   public String getGuestNames() {
      String str = new String();
      for( Guest existing : guests )
         str += existing.toString() + "\t";
      return str;

   }
   public Visit getVisit() { //to report current revenue
      return v;
   } //or add a method which calculates the revenue
}
```

Question 4: Write the source code of the classes ReportCurrentRevenue and ReportPastIncome. (25p)

```
import java.util.*;
public class ReportCurrentRevenue
implements Runnable {
   private Hashtable<Integer,Room> rooms;
   public ReportCurrentRevenue(
       Hashtable<Integer,Room> rooms ) {
       this.rooms = rooms; }
   public void run() {
      double amount = 0.0;
      for( Room room : rooms.values() ) {
         Visit v = room.getVisit();
         amount += v.getCurrentRevenue();
      }
      System.out.println("Revenue:" +
      amount);
   }
}
```

```
public class ReportPastIncome implements
Runnable {
   private LinkedList<Visit> visits;
   public ReportPastIncome(
      LinkedList<Visit> visits) {
      this.visits = visits; }
   public void run() {
      double amount = 0.0;
      for( Visit v : visits ) {
         amount += v.getTotalPrice();
      }
      System.out.println("Past income:" +
      amount);
   }
}
```

Question 5: Write the source code of only the following methods of the class Hotel. (25p)

```java
public void report() {
  pool = Executors.newCachedThreadPool( );
  pool.execute( new ReportCurrentRevenue(rooms) );
  pool.execute( new ReportPastIncome(visits) );
  pool.shutdown( );
}
public void restore( String fileName ) {
  while( pool.isTerminated() );
  try {
    ObjectInputStream str = new ObjectInputStream(
         new FileInputStream(fileName));
    rooms = (Hashtable<Integer,Room>)str.readObject();
    visits = (LinkedList<Visit>)str.readObject();
    str.close();
  }
  catch(ClassNotFoundException e) { e.printStackTrace(); }
  catch(IOException e) { e.printStackTrace(); }
}
public int findRoomNrOfGuest( String name ) {
  for( Room room : rooms.values() ) {
    Guest guest = room.searchGuest(name);
    if( guest != null )
      return room.getRoomNr();
  }
  return 0;
}
```