BLM2562 OBJECT ORIENTED PROG., Section 2, 1ST MIDTERM (06/04/2015)

Question 1: Write the source code of class Doctor. (15p)

```java
public class Doctor extends Person {
   private static final long serialVersionUID = 1L;
   private int diploma_id;
   private Schedule schedule;

   public Doctor(String name, long national_id, int diploma_id) {
      super(name, national_id);
      this.diploma_id = diploma_id;
   }
   public int getDiploma_id() { return diploma_id; }
   public Schedule getSchedule() { return schedule; }
   public void setSchedule(Schedule schedule) { this.schedule = schedule; }
   public String toString() {
      return super.toString() + " (Diploma ID: " + diploma_id + ")";
   }
}
```

Question 2: Write the source code of class DuplicateInfoException. (10p)

```java
@SuppressWarnings("serial")
public class DuplicateInfoException extends RuntimeException {
   public DuplicateInfoException( String msg ) {
      super(msg);
   }
}
```

Question 3: Write the source code of class Section. (20p)

```java
import java.util.*;
public class Section implements java.io.Serializable {
   private static final long serialVersionUID = 1L;
   private final int id;
   private String name;
   private LinkedList<Doctor> doctors;

   public Section(int id, String name ) {
      this.id = id;
      this.name = name;
      doctors = new LinkedList<Doctor>();
   }
   public Doctor getDoctor( int id ) {
      for( Doctor aDoctor : doctors )
         if( aDoctor.getDiploma_id() == id )
            return aDoctor;
      return null;
   }
   public void listDoctors() {
      for( Doctor aDoctor : doctors )
         System.out.println(aDoctor);
   }
   public void addDoctor( Doctor doctor ) {
      if( getDoctor(doctor.getDiploma_id()) != null )
         throw new DuplicateInfoException(
               "A doctor with the same diploma ID already exists"
               + ". ID: " + doctor.getDiploma_id());
      doctors.add(doctor);
   }
   public int getId() { return id; }
   public String getName() { return name; }
}
```

Question 4: Write the source code of the addRendezvous method of class Schedule. (20p)

```
public boolean addRendezvous( Patient p, Date desired ) {
   int rendezvousCount = 0;
   Calendar wanted = Calendar.getInstance();
   wanted.setTime(desired);
   for( Rendezvous rand : sessions ) {
      Calendar current = Calendar.getInstance();
      current.setTime(rand.getDateTime());
      if( wanted.get(Calendar.YEAR) == current.get(Calendar.YEAR) &&
            wanted.get(Calendar.DAY_OF_YEAR) == current.get(Calendar.DAY_OF_YEAR) )
         rendezvousCount++;
   }
   if( rendezvousCount < maxPatientPerDay ) {
      sessions.add(new Rendezvous(p, doctor, desired));
      return true;
   }
   return false;
}
```

Question 5: Write the source code of makeRandezvous method of class CRS. (20p)

```
public boolean makeRendezvous( long patientID, int hospitalID,
      int sectionID, int diplomaID, Date desiredDate ) {
   Patient patient = patients.get(patientID);
   if( patient == null )
      throw new IDException("Bad ID for a patient: " +  patientID );
   Hospital hospital = hospitals.get(hospitalID);
   if( hospital == null )
      throw new IDException("Bad ID for a hospital: " +  hospitalID );
   Section section = hospital.getSection(sectionID);
   if( section == null )
      throw new IDException("Bad ID for a section: " +  sectionID );
   Doctor doctor = section.getDoctor(diplomaID);
   if( doctor == null )
      throw new IDException("Bad ID for a doctor: " +  diplomaID );
   Schedule schedule = doctor.getSchedule();
   Schedule schedule = doctor.getSchedule();
   if( schedule == null ) {
      System.out.println("The doctor does not have a schedule yet!");
      return false;
   }
   boolean result = schedule.addRendezvous(patient, desiredDate);
   if( result ) {
      Rendezvous r = new Rendezvous(patient, doctor, desiredDate);
      rendezvous.add(r);
   }
   return result;
}
```

Question 6: Write the source code of saveTablesToDisk method of class CRS. (15p)

```
public void saveTablesToDisk( String fullPath ) {
   try {
      ObjectOutputStream output = new ObjectOutputStream(
            new FileOutputStream(fullPath) );
      output.writeObject(patients); output.writeObject(rendezvous);
      output.writeObject(hospitals);
      output.close();
   }
   catch (IOException e) {
      System.out.println("Problem while saving tables.");
      e.printStackTrace();
   }
}
```