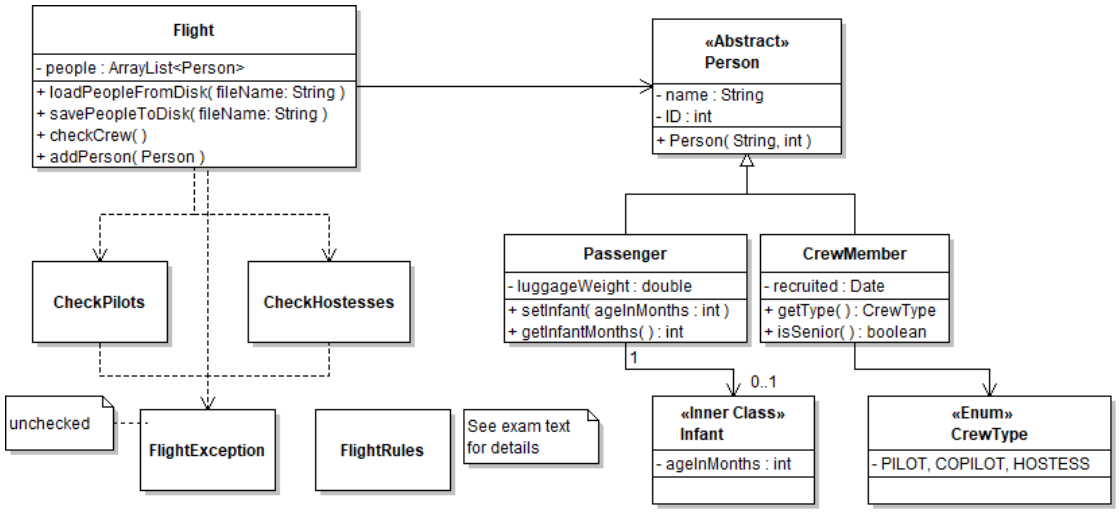
YILDIZ TEKNİK ÜNİVERSİTESİ

ELEKTRİK-ELEKTRONİK FAKÜLTESİ / BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Öğrencinin Adı Soyadı: Rayene Bech	**Öğrenci No: 18011115**			**İmza:**	
Dersin Adı: BLM2012 Nesneye Yönelik Progr.	**Tarih/Saat:** 17/06/2021 11:00			**Sınav süresi:** 90dk	
Sınav Türü:	**Vize1**	**Vize2**	**Mazeret**	**Final +**	**Bütünleme**
Unvan Ad Soyad (Ders Yürütücüsü): **Gr.1** Doç.Dr. Mehmet S. Aktaş **Gr.2** Dr.Öğr.Üyesi Yunus E. Selçuk **Gr.3** Öğr.Gör.Dr. Ahmet Elbir					

QUESTIONS




```
public class FlightRules {
	public final static int minHostessCount = 2;
	public final static int minSeniorHostessCount = 1;
	public final static int yearsToBecomeSenior = 3;
}
```

Not: Bu belgenin sonunda ÖNEMLİ NOTLAR bölümünde verilen kurallara uygun olarak cevaplarınızı hazırlayınız ve sisteme yükleyiniz.

Answer these questions according to the UML class schema and the code given above. You may need to extract hidden information from the schema and add necessary code while answering the questions.

Question 1 (10 pts): Write the source code of enum CrewType.

```
enum CrewType{
    PILOT, COPILOT, HOSTESS;
 }
```

Question 2 (15 pts): Write the source code of the class CrewMember. A crew member cannot change its duty type. For example, a hostess can never become a pilot. The number of years to become a senior member should be obtained from the class FlightRules.

```
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
public class CrewMember extends Person {

   private Date recruited;
   private  CrewType type;

   public CrewMember(String name, int id, Date date, CrewType type) {
```

```
        super(name, id);
        this.recruited = date;
        this.type = type;
    }

    public Date getRecruited() {
        return recruited;
    }

    public CrewType getType() {
        return type;
    }
    public boolean isSenior(){
       Date currentdate = new Date(System.currentTimeMillis());
        long diffInYears = currentdate.getTime() - this.recruited.getTime() ;
        int duration = (int)(diffInYears / (1000l * 60 * 60 * 24 * 365)) ;
        if(duration > FlightRules.yearsToBecomeSenior){
            return true;
        }
        return false;
    }

}
```

Question 3 (15 pts): Write the source code of the class Passenger. A passenger can travel with an infant (i.e. his/her child). Code the class Infant as an inner class of the class Passenger. If a passenger is not travelling with an infant, the getInfantMonths method must return –1.

```
public class Passenger extends Person {
    private double luggageWeight;
    Infant anInfant;
    public Passenger(String name, int id, double luggageWeight) {
        super(name, id);
        this.luggageWeight = luggageWeight;
    }

    public void setInfantMonths(int infantMonths) {
      anInfant= new Infant(infantMonths);

    }

    public double getLuggageWeight() {
```

```
        return luggageWeight;
    }

    public int getInfantMonths() {
        if(anInfant == null){
            return -1;

        }
        else{
            return anInfant.getAgeInMonths();
        }
    }




    class Infant{
        private int ageInMonths;

        public Infant(int ageInMonths) {
            this.ageInMonths = ageInMonths;
        }

        public int getAgeInMonths() {
            return ageInMonths;
        }

        public void setAgeInMonths(int ageInMonths) {
            this.ageInMonths = ageInMonths;
        }

    }
}
```

Question 4 (10 pts): Write the source code of the exception FlightException.

```
public class FlightException extends  RuntimeException {
    public FlightException (String message){
        super(message);
    }
    public  FlightException (){
        super();
```

```
    }
}
```

Question 5 (15 pts): Write the source code of the multithreaded class of CheckPilots. The details of this class is given in Question 6 as an instance of this class is used in the Flight.checkCrew() method.

```
import java.util.ArrayList;

public class CheckPilots implements Runnable{
    private ArrayList<Person> people;
    public CheckPilots(ArrayList<Person> people){
        this.people = people;
    }
    public void run() {
        boolean pilotfound= false, copilotfound = false;
        for(Person aperson: people){
            if (aperson instanceof CrewMember){
                if(((CrewMember)aperson).getType() == CrewType.PILOT){
                    pilotfound = true;
                }
            }
            if (aperson instanceof CrewMember){
                if(((CrewMember)aperson).getType() == CrewType.COPILOT){
                    copilotfound = true;
                }
            }
        }
        if(pilotfound && copilotfound ){
            System.out.println("The flight has a pilot an copilot");
        }
        else if(pilotfound){
            System.out.println("Checking failed! the flight does not have a copilot ");
        }
        else if(copilotfound){
            System.out.println("Checking failed! the flight does not have a polit ");
        }
        else{
            System.out.println("Checking failed! the flight does not have a copilot nor a polit");
        }

    }
```

}

Question 6 (35 pts): Write the source code of the class Flight. Some details of this class is as follows:

- addPerson: This method creates a FlightException if a person with the same ID is attempted to add to the people list multiple times.
- checkCrew: This method is responsible from creating an instance of CheckHostesses and an instance of CheckPilots in separate threads and from executing them. The CheckHostesses instance checks the people list for hostesses. A flight must contain at least two hostesses and at least one of them must be senior worker. The CheckPilots instance checks the people list for pilots because a flight must contain a pilot and a copilot.
- loadPeopleFromDisk and savePeopleToDisk: If a crew checking operation is in progress, those methods must wait for the end of that checking operation.

```
import java.io.EOFException;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.ArrayList;



public class Flight {
   private ArrayList<Person> people;

   public Flight() {
      people = new ArrayList<Person>();
   }




   public synchronized void  loadPeopleFromDisk(String FileName){
      try{
         FileInputStream file =  new FileInputStream(FileName);
         ObjectInputStream inputFile = new ObjectInputStream(file);
         boolean EOF = false;
         while(!EOF){
            try{
               people.add( (Person)inputFile.readObject());
            }
            catch(EOFException e){
               EOF = true;
```

```
            }
            catch(Exception f){

            }
        }
        inputFile.close();

    }
    catch(IOException e){
      e.printStackTrace();
    }
}

public synchronized void  savePeopleToDisk(String FileName){
     try{
        FileOutputStream file= new FileOutputStream(FileName);
        ObjectOutputStream outputFile =  new ObjectOutputStream(file);

        for(int i=0; i< people.size(); i++){
            outputFile.writeObject(people.get(i));
        }

        outputFile.close();

    }
    catch(IOException e){
        e.printStackTrace();

    }
}


public void checkCrew() throws InterruptedException{
    Runnable r1 =  new CheckPilots(people);
    Thread t1 = new Thread (r1);
    Runnable r2 =  new CheckHostesses(people);
    Thread t2 = new Thread (r1);
    t1.start();
    t2.start();
    t1.join();
    t2.join();
```

```
    }
    public void addPerson(Person aperson) throws FlightException{
        for(Person person: people){
            if(person.getId() == aperson.getId()){
                throw new FlightException("A person with the same ID already exists");
            }
        }
        people.add(aperson);
    }
}
```

ÖNEMLİ NOTLAR

- Çözümlerinizin tamamını sınav sorularının düzenindeki şekliyle sadece bir WORD dosyası haliyle göndermeniz gerekmektedir.
- IDE kullanabilirsiniz. Cevaplarınızı sınav sorularını içeren dosyaya yazınız.
- Sınavın son 5-10 dakikasını cevap yükleme süresi olarak ayırmayı unutmayınız. Kendiniz saat tutunuz, sistemin gösterdiği saat içsel olarak doğru olsa da browser'da düzgün tazelenmeyebiliyor.
- Yüklemenin başarılı olduğunu gösteren ekranın ekran görüntüsünü delil olarak kaydediniz. Delil olmadan, olağan dışı hatalar sorunlar olunca diğer kanallardan gönderilen çözümler kabul edilmeyecektir.