

Zusatzaufgaben OMP – Code modellieren

Gegeben sei der folgende Code. Ihre Aufgabe ist es, diesen in einem UML-Klassendiagramm zu modellieren.

```
public class Hospital {
    private String name;
    private ArrayList<Physician> physicians = new ArrayList<>();
    private ArrayList<Patient> patients = new ArrayList<>();
    private ArrayList<Medicine> medicines = new ArrayList<>();

    public Hospital(){}

    public ArrayList<Physician> getPhysicians() {
        return physicians;
    }

    public ArrayList<Patient> getPatients() {
        return patients;
    }

    public ArrayList<Medicine> getMedicines() {
        return medicines;
    }

    public void addPhysician(Physician physician){
        physicians.add(physician);
    }

    public void addPatient(Patient patient){
        if(patient.treatable(medicines)){
            patients.add(patient);
        }
    }

    public void addMedicin(Medicine medicine){
        medicines.add(medicine);
    }

    public boolean treatPatient(int index, Treatment treatment){
        if(patients.size()<=index){

        }else if(patients.get(index).treat(treatment)) {
            patients.remove(index);
            return true;
        }
        return false;
    }

    public boolean fireStaff(Physician staffMember){
        if(physicians.contains(staffMember)){
            return physicians.remove(staffMember);
        }
        return false;
    }

    public String getName() {
        return name;
    }
}
```

```

    public void setName(String name) {
        this.name = name;
    }

}

public class Physician {
    private String name;
    private Hospital workplace;
    private ArrayList<Patient> patients = new ArrayList<>();

    public Physician(Hospital workplace){
        this.workplace = workplace;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void changeWorkplace(Hospital newWorkplace){
        if(newWorkplace != null && !workplace.equals(newWorkplace)){
            workplace.fireStaff(this);
            workplace = newWorkplace;
            workplace.addPhysician(this);
        }
    }

    public void loseJob(){
        workplace.fireStaff(this);
        workplace = null;
    }

    public void getHired(Hospital workplace){
        this.workplace = workplace;
    }

    public void addPatient(Patient patient){
        if(workplace.getPatients().contains(patient)){
            patients.add(patient);
        }
    }
}

public class Patient {
    private Disease disease;
    private String name;

    public Patient(Disease disease){
        this.disease = disease;
    }

    public Disease getDisease() {
        return disease;
    }
}

```

```

    public boolean treatable(List<Medicine> medicines){
        return disease.treatable(medicines);
    }

    public boolean treat(Treatment treatment){
        if(treatment.equals(disease.getTreatment())){
            disease = null;
            return true;
        }
        return false;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}

public class Disease {
    private String name;
    private Treatment treatment;

    public Disease(String name){
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public Treatment getTreatment() {
        return treatment;
    }

    public void setTreatment(Treatment treatment) {
        this.treatment = treatment;
    }

    public boolean treatable(List<Medicine> medicines){
        return treatment.possible(medicines);
    }
}

public class Treatment {
    private String name;
    private ArrayList<Medicine> medicines = new ArrayList<>();

    public Treatment(String name, ArrayList<Medicine> medicines){
        this.name = name;
        this.medicines = medicines;
    }

    public String getName() {
        return name;
    }

    public ArrayList<Medicine> getMedicines() {
        return medicines;
    }
}

```

```
public void setMedicines(ArrayList<Medicine> medicines) {
    this.medicines = medicines;
}

public boolean possible(List<Medicine> medicines){
    for(Medicine medicine: this.medicines){
        if(!medicines.contains(medicine)){
            return false;
        }
    }
    return true;
}

}

public class Medicine {
    private String name;

    public Medicine(String name){
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```