Surgery Planner

Einleitung

Für ein Spital soll ein System entwickelt werden um Behandlungen (Surgeries) zu verwalten. Die Hauptlogik wird in der Klasse `SurgeryPlanner` abgebildet. Diese bietet Methoden an um Behanldungen zu registrieren um dieselben dann in entsprechenden Listen den Anwender:innen zur Verfügung stellen zu können.

Modell und Implementierung

Abb zeigt das Klassendiagramm der beteiligten Klassen.

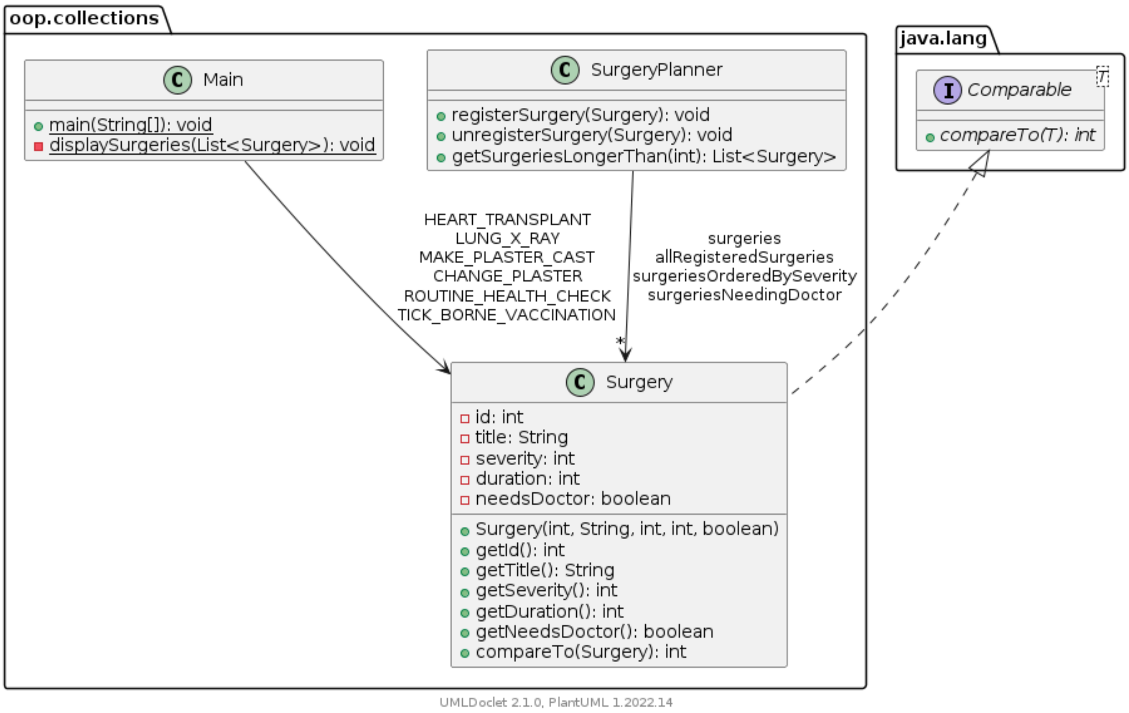

Figure 1: Klassendiagramm des Surgery Planner Projektes

Die Klasse `Surgery` bildet eine Behandlung ab. Die Klasse `SurgeryPlanner` implementiert die benötigten Operationen (Methoden) zur Verwaltung der Behandlungen.

`Main` demonstriert die Verwendung

```java
public class Main {

    public static final Surgery HEART_TRANSPLANT =
        new Surgery(4, "Heart transplant", 4, 120, true);
    public static final Surgery LUNG_X_RAY =
        new Surgery(5, "Lung x-ray", 3, 45, false);
    public static final Surgery MAKE_PLASTER_CAST =
        new Surgery(1, "Make plaster cast", 2, 15, true);
    public static final Surgery CHANGE_PLASTER =
        new Surgery(2, "Change Plaster", 1, 10, false);
    public static final Surgery ROUTINE_HEALTH_CHECK =
        new Surgery(3, "Routine health check", 1, 30, true);
    public static final Surgery TICK_BORNE_VACCINATION =
        new Surgery(6, "Tick borne vaccination", 1, 10, false);

    public static void main(String[] args) {
        SurgeryPlanner planHospitalSurgeries = new SurgeryPlanner();
        planHospitalSurgeries.registerSurgery(MAKE_PLASTER_CAST);
        planHospitalSurgeries.registerSurgery(CHANGE_PLASTER);
        planHospitalSurgeries.registerSurgery(ROUTINE_HEALTH_CHECK);
        planHospitalSurgeries.registerSurgery(HEART_TRANSPLANT);
        planHospitalSurgeries.registerSurgery(LUNG_X_RAY);
        planHospitalSurgeries.registerSurgery(TICK_BORNE_VACCINATION);

        System.out.println("All registered surgeries !");
        displaySurgeries(planHospitalSurgeries.getAllRegisteredSurgeries());
        System.out.println("All surgeries that take longer than 10 minutes !");
        displaySurgeries(planHospitalSurgeries.getSurgeriesLongerThan(10));
        System.out.println("All surgeries that require a doctor !");
        displaySurgeries(planHospitalSurgeries.getSurgeriesNeedingDoctor());
        System.out.println("Surgeries ordered by severity starting with the less severe ones !");
        displaySurgeries(planHospitalSurgeries.getSurgeriesOrderedBySeverity());
    }

    private static void displaySurgeries(List<Surgery> surgeries) {
        for (Surgery current : surgeries) {
            System.out.println(current);
        }
    }
```

Aufgabe

Implementieren Sie die beiden Klassen `Surgery` und `SurgeryPlanner` .