

OPDRACHT: PANDEMICSIMULATOR

Je begint als verpleegkundige bij de nieuwe hospitalisatie branche van Intec.

Je komt af op je eerste dag, en je ziet een overvolle ruimte vol met zieke mensen, en gestresseerde collega's.

Al hun problemen kun je oplossen met je huidige kennis.

Vereisten:

Creëer de klasse 'Patient' met als variabelen: nationalRegistryNumber(String), fullName(String), age(int), temperature(int), unknownVirus(boolean), ensured(boolean).

Zorg voor de nodige getters en setters, equals en hashCode en een toString methode.

Plak volgende methode in je Patient klasse:

```
public static List <Patient> getAllPatients(){
    List <Patient> allPatients = new ArrayList<>();

    Patient donaldPutin = new Patient("110201321", "Donald Putin", 90, 39, false, true);
    allPatients.add(new Patient("030201321", "Lucky Luke", 69, 38, true, false));
    allPatients.add(new Patient("030201322", "Stromae Papaoutai", 35, 40, false, true));
    allPatients.add(new Patient("030801321", "Gringoire Lune", 80, 36, true, true));
    allPatients.add(new Patient("687201321", "Crazy Frog", 5, 39, false, false));
    allPatients.add(new Patient("030244321", "Sanic the HedgeHog", 16, 37, true, true));
    allPatients.add(new Patient("110201321", "Donald Putin", 90, 39, false, true));
    allPatients.add(new Patient("110201321", "Donald Putin", 90, 39, false, true));
    allPatients.add(new Patient("088201321", "Bilbo Baggins", 23, 39, true, false));
    allPatients.add(new Patient("035501321", "Lucio Correa Dos Santos", 40, 38, false, true));
    allPatients.add(new Patient("030206621", "Doctor Who", 33, 35, true, true));
    allPatients.add(new Patient("030245321", "Luke Cage", 46, 41, true, false));
    allPatients.add(new Patient("030298321", "Naruto Uzumaki", 13, 38, false, false));
    allPatients.add(new Patient("125201321", "Billy Batson", 15, 40, true, false));
    allPatients.add(new Patient("987201321", "Joseph Joestar", 75, 36, true, false));
    allPatients.add(new Patient("645201321", "Jon Snow", 33, 35, false, false));
    allPatients.add(new Patient("368201321", "Prince Charming", 95, 37, true, true));
    allPatients.add(new Patient("582201321", "Alexander Hamilton", 65, 38, false, false));
    allPatients.add(new Patient("687201321", "Crazy Frog", 5, 39, false, false));
    allPatients.add(new Patient("030201231", "Zelda of Hyrule", 17, 38, false, true));
    allPatients.add(new Patient("830208651", "Mother Theresa", 98, 39, true, false));
```

```

allPatients.add(new Patient("030209741", "Willy Wonka", 70, 40, true, true));
allPatients.add(new Patient("030205411", "Scrooge McDuck", 73, 38, true, false));
allPatients.add(new Patient("368201321", "Prince Charming", 95, 37, true, true));

allPatients.add(new Patient("030209861", "Sonia Magnlia", 35, 39, false, false));
allPatients.add(new Patient("030278921", "Peter Parker", 21, 40, true, false));
allPatients.add(new Patient("030269821", "Bruce Wayne", 46, 35, false, true));
allPatients.add(new Patient("030215421", "Seto Kaiba", 22, 38, false, true));
allPatients.add(new Patient("030296321", "Mario Mario", 50, 37, false, false));
allPatients.add(new Patient("035051321", "Harry Potter", 18, 39, true, true));
allPatients.add(new Patient("368201321", "Prince Charming", 95, 37, true, true));
allPatients.add(new Patient("036781321", "Mary Poppins", 38, 38, true, false));
allPatients.add(new Patient("033691321", "Alexander the Great", 51, 37, false, true));
allPatients.add(new Patient("687201321", "Crazy Frog", 5, 39, false, false));
allPatients.add(new Patient("030201000", "Huey Freeman", 12, 40, false, false));
allPatients.add(new Patient("000201321", "Miku Hatsune", 16, 36, true, true));
allPatients.add(new Patient("111201321", "Drake Josh", 33, 38, true, false));
allPatients.add(new Patient("222201321", "April O'Neil", 29, 37, false, false));
return allPatients;
}

```

Opdracht 1 (/1):

De hoofdverpleegster wilt dat je alle patienten in een lijst stopt die ervoor zorgt dat elke Patient er maar 1 keer in voorkomt, liefst op volgorde dat ze binnen kwamen.

---Zorg ervoor dat je een Collection gebruikt die alleen unieke elementen weergeeft, en alles in volgorde houdt. Print de lijst af.

Opdracht 2 (/3):

Terwijl dat je de lijst aan het opmaken ben, krijgen jullie een noodbericht van de regering: er is een nieuw onbekend virus uitgebroken!

Het lijkt erop dat de regering vraagt om voorrang te verlenen aan patiënten die een hoge temperatuur hebben, en dan aan patiënten met een hogere leeftijd.

Bonus (/1):

Je baas merkt op dat dit hospitaal werkt op een Amerikaans systeem: als patiënten een verzekering hebben, zou zij het wel leuk vinden als je deze ook zou willen voor trekken na rekening te hebben gehouden met de directieven van de regering.

---Sorteer een List van patiënten aan de hand van TemperatureSorter en AgeSorter die de klasse Comparator<Patient> implementeren. Optioneel make je een InsuranceSorter. Je mag hier ook lambdas/streams voor gebruiken!

Giet de resultaten achteraf in een Queue. Print de lijst af.

Opdracht 3 (/4):

De regering geeft als laatste opdracht om patiënten van elkaar te isoleren, want ze mogen elkaar niet besmetten. Het systeem dat ze willen gebruiken gaat als volgt:

Category 1: Iedereen dat hoge koorts heeft (≥ 40) of bejaarden (≥ 65) met gewone koorts of hoger (≥ 38), op voorwaarde dat ze daarboven het onbekende virus hebben. `((patient.getAge<=65 && patient.getTemperature >=38) || patient.getTemperature()>=40) && patient.isUnknownVirus()`

Category 2: De rest van de bevolking die met gewone koorts of hoger (≥ 38) loopt, en het onbekende virus in hun hebben.

Category 3: Mensen die het onbekende virus hebben, maar geen koorts.

Category 4: Mensen die koorts hebben (≥ 38) maar met een bekend virus rondlopen. Overige patiënten worden naar huis gestuurd of naar de apotheker.

---Gebruik een Map die als categorieën 1 t.e.m. 4 als Key gebruikt. The value is telkens een List die je opvult met de patiënten in de Queue.

Print zowel de resultaten van de Map af, als de inhoud van de Queue, die leeg moet zijn.
Je kunt het volgende gebruiken voor over je Map te itereren.

```
for(Map.Entry<Integer, List<Patient>> category : mapCollection.entrySet()){  
    int key = category.getKey();  
    //use a method to iterate over your patientsList}
```

---SUCCES---