Aufgabe 2 – Priority Queue

Vorbereitung

- Machen Sie sich anhand der Unterlagen aus der Vorlesung mit der Datenstruktur des *HEAP* vertraut.
- Überlegen Sie, durch welche Änderungen aus dem *MAXHEAP* aus der Vorlesung ein *MINHEAP* gemacht werden kann.

Aufgabenbeschreibung

1. Entwickeln Sie die Java Klasse **`PQElement`** entsprechend dem folgenden Muster:

```
class PQElement {
    private String data;
    private double priority;

    public PQElement(String s, double d) { ... }
    public String getData() { ... }
    public double getPrio() { ... }
    public void setPrio(double d) { ... }
}
```

   Entwickleln Sie für diese Elementklasse die Java Klasse **`MinPQ`** die eine Priority Queue in Form eines *MINHEAP* implementiert. Verwenden Sie dafür folgendes Muster:

```
public class MinPQ {
    private PQElement[];
    private int maxsize;
    private int currentsize;

    public MinPQ(int max) { ... } // Konstruktor
    public boolean isEmpty() { ... }
    public boolean insert(PQElement n) { ... }
    public boolean insert(String d, double p) { ... }
    public PQElement extractElement() { ... }
    public String extractData() { ... }
}
```

   Die Methode **`isEmpty()`** überprüft, ob die Priority Queue leer ist.
   Die beiden Methoden **`insert()`** fügen ein neues Element in die Priority Queue ein.
   Die Methode **`extractElement()`** entfernt das Element mit der geringsten Prioriät aus der Priority Queue und liefert das Element an den Anufrufer.
   Die Methode **`extractData()`** entfernt das Element mit der geringsten Priorität und liefert den Datenstring an den Aufrufer.

Schreiben Sie eine Java Programm, dass diese Klasse testet, indem 1000 Elemente mit Zufallsprioritäten in die Priority Queue eingefügt werden und anschließend der Reihe nach wieder aus der Priority Queue entfernt werden. Überprüfen Sie, ob die Ordnung der Elemente korrekt ist.

Verwenden Sie zur Erzeugung der Pseudozufallszahlen die Klasse **`Java.util.Random`**
(siehe https://docs.oracle.com/javase/8/docs/api/java/util/Random.html).

2. Erweitern Sie die Klasse **`MinPQ`** durch die Methode:

```
public void update(String s, double n) { ... }
```

Diese Methode ändert den Prioritätswert des Elementes mit dem Datenstring **`s`** auf den neuen Wert **`n`**. Der Index des Elementes in der Priority Queue kann durch sequentielle Suche ermittelt werden. Beachten Sie, dass der Prioritätswert **nur kleiner** als der bisherige sein soll.

Testen Sie diese Methode indem Sie in den 1000 Elementen mit Zufallsprioritäten für jedes zweite Element die Priorität verringern. Überprüfen Sie, ob die Ordnung noch korrekt ist.

Abgabe

Laden Sie das Programm ihrer Gruppe im OSCA-Abgabebereich der Lehrveranstaltung für das Praktikum hoch. Bilden Sie dazu aus Ihrem Projekt ein ZIP-Archiv, dass nur die Quelldateien enthält. Der Name des Archivs muss die Namen der Gruppenmitglieder enthalten, also z.B. **`Meyer_Schulze.zip`**. Dieses Archiv muss zu Beginn Ihres nächsten Praktikumstermins in OSCA vorliegen.

Bitte beachten Sie die allgemeinen Hinweise zum Praktikum.