

Name: _____

Tutor: _____

Matrikelnummer: _____

Punkte: _____

Gruppe: _____

Abgabe: Di, 24. 5. 2005

1 Moderne Behörde (14 Punkte)

Ein Ministerium soll modernisiert und die gesamte Verwaltung auf EDV umgestellt werden. Darunter fällt auch die Verwaltung von eingehenden Anträgen in einer Warteschlange, bis sie anschließend zur Bearbeitung an einen Beamten weitergeleitet wird.

Die Verwaltung dieser Anträge war früher ein großes Problem, da sich in dieser Warteschlange oftmals sehr viele Anträge befanden. Die zuständigen Beamten mussten daher immer lange suchen, um den Antrag mit der höchsten Priorität (angegeben in Euro) zu finden und diesen dann zu bearbeiten. Anträge mit gleicher "Priorität" wurden in der Reihenfolge der Antragstellung erledigt.

Implementieren Sie diese Prioritätswarteschlange als Heap, um die Verwaltung zu beschleunigen:

```
public class AdministrationQueue {
    public AdministrationQueue() { ... }
    public void insert(Request r, int priority) { ... }
    public Request next() { ... }
    public Request peek() { ... }
    public boolean hasNext() { ... }
    public int getQueueLength() { ... }
    public void printAll() { ... }
}
```

Implementieren Sie auch einen entsprechenden Testtreiber.

Die Priorität soll nicht im Request abgelegt werden.

Implementieren Sie den Heap keinesfalls als Baum (Das ist nicht Sinn der Sache). Sie können dafür aber z.B. die Klasse `java.util.ArrayList` verwenden.

2 Graphen (10 Punkte)

Gegeben seien folgende Datenstrukturen:

- 2D-Suchbaum (wie bei Übung 3).
- Ein Heap (wie in Aufgabe 1).
- Modell des öffentlichen Verkehrs in Linz zur Simulation der Streckenauslastung.
- Modell des Kanalsystems in Linz.
- Modell des Eisenbahnnetzes für Überlandverbindungen in der EU (d.h. Bahnhöfe können jeweils als ein Knoten modelliert werden)
- Modell einer Platine für Radios (ein Bauteil als ein Knoten modelliert).

Stellen diese Datenstrukturen (im Modell) Graphen dar? Wenn nein, warum nicht?

Wenn ja, kategorisieren Sie diese Graphen nach gerichteter/ungerichteter, gewichteter/ungewichteter, vollständiger/nicht vollständiger, zyklischer/azyklischer und zusammenhängender/unzusammenhängender Graph. Begründen Sie jeweils Ihre Entscheidungen.

Hinweis: Es ist nicht so wichtig, wie Sie die einzelnen Datenstrukturen kategorisieren (nicht immer eindeutig, manchmal von der Sichtweise und der Detailtreue des Modells abhängig). Viel wichtiger ist die genaue Begründung für Ihre Kategorisierung.