9. Übungsblatt „Programmiersprachen“ SS 99

Leif Kornstaedt, Prof. Gert Smolka

Ausgabe: 18. Juni 1999 **Abgabe:** 25. Juni 1999

Aufgabe IX.1: Schlangen in Java (10 Punkte)

Implementieren Sie Schlangen wie folgt:

```
class EmptyQueueException extends Exception {};

class Queue {
    public Queue();
    public void enqueue(Object o);
    public Object dequeue() throws EmptyQueueException;
    public boolean isEmpty();
};
```

Alle Operationen sollen konstante Laufzeit haben.

Aufgabe IX.2: Tabellen in Java ($15 + 8 + 10$ Punkte)

a. Implementieren Sie Tabellen durch binäre Suchbäume wie folgt:

```
interface Comparable {
    public int compareTo(Object o);
};

class Table {
    public Table();
    public void insert(Comparable key, Object entry);
    public Object lookup(Comparable key);
};
```

Dabei soll die Operation `lookup` in dem Fall, dass der Schlüssel nicht in der Tabelle enthalten ist, `null` zurückliefern.

Bemerkung: Das Interface `Comparable` ist erst ab der Version 1.2 in der Java API enthalten. Sie können die Aufgabe aber auch mit Java 1.1 lösen.

b. Schreiben Sie ein ausführbares Programm `TestTable`, das eine Tabelle mit folgendem Inhalt erzeugt:

```
"adalbert"     1
"edith"        5
"marie-luise"  17
"dorothea"     8
```

Das Programm erhält auf der Kommandozeile eine Zeichenkette und schlägt diese in der Tabelle nach. Existiert der Eintrag, so wird er ausgegeben, sonst eine entsprechende Meldung.

Beispiel:

```
java TestTable x
```

erzeugt die Ausgabe

```
key x not found
```

Bei ungültigem Aufruf soll eine Fehlermeldung ausgegeben werden.

*c. Wir erweitern die Klasse `Table` um eine Operation, die eine Aufzählung aller Schlüssel erzeugt:

```
public java.util.Enumeration keys();
```

Implementieren Sie `keys`. Erweitern Sie `TestTable` so, daß es alle Schlüssel der erzeugten Tabelle ausgibt, wenn es ohne Parameter aufgerufen wird.