Prof. G. Zachmann
Daniel Mohr

TU Clausthal
Institut für Informatik
28. April 2011

Sommersemester 2011

# Übungen zu Informatik II - Blatt 3

Abgabe am 05.05

**Organisatorisches**

- Die theoretischen Aufgaben müssen Sie donnerstags in der Vorlesung abgeben.
- Die Programmieraufgaben müssen Sie donnerstags bis spätestens 13:15 Uhr an Ihren Tutor per Email (**christian.schnarr@tu-clausthal.de**) schicken.
- Die Programmieraufgaben müssen von Ihnen in der Übung vorgeführt und erklärt werden.

## Aufgabe 1 (Doppelt verkettete Liste, *2 Punkte*)

Gegeben ist eine doppelt verkettete Liste, deren Listenelemente so definiert sind:

```
class ListElement2(object):

    def __init__( self ):
        self.item = self.next = self.prev = None
```

Gegeben ist außerdem wieder die Variable `cursor`, wie in der Vorlesung besprochen.

Gegeben ist nun ein Element `z`, d.h., dieses wurde erzeugt mit `z = ListElement2()`.

Schreiben Sie die Listenfunktion `insertAfterCursor(z)` in Python, mit der `z` *nach* dem Listenelement, auf das `cursor` zeigt, in die Liste eingefügt wird. (Denken Sie auch an die Spezialfälle!)

## Aufgabe 2 (Multi-Listen, *7 Punkte*)

Implementieren Sie eine Klasse `SparseMatrix` zur Repräsentation quadratischer, dünn besetzter Matrizen. Verwenden Sie dazu die aus der Vorlesung bekannten Multi-Listen. Die Klasse soll mindestens folgende Funktionen haben:

- Dem Kontruktor soll die Anzahl der Spalten und Zeilen übergeben werden.
- Implementieren Sie eine Methode `set()` zum Setzen der Matrixeinträge.
- Implementieren Sie eine Methode `get( )` die den Wert eines Matrixeintrags zurückliefert.
- Schreiben Sie eine Funktion (Methode) zur Addition zweier solcher Matrizen
- Schreiben Sie eine Funktion (Methode) zur Multiplikation zweier solcher Matrizen
- Entwerfen Sie eine Funktion `transpose`, die die transponierte Matrix zurück liefert (d.h. die Einträge der Spalten und Zeilen werden vertauscht) .

- Implementieren Sie eine Funktion **print** zur Ausgabe der Matrix.

Halten Sie sich dabei bitte an folgende Deklarationen:

```
class SparseMatrix(Object):
    def __init__ (self, rows, cols):
        # hier kommt Ihr Code
    def set(self, row, col, value):
        # hier kommt Ihr Code
    def get(self, row, col):
        # hier kommt Ihr Code
        return #...
    def add(self, other_matrix):
        new_matrix = SparseMatrix()
        # hier kommt Ihr Code
        return new_matrix
    def mul(self, other_matrix):
        new_matrix = SparseMatrix()
        # hier kommt Ihr Code
        return new_matrix
    def transpose(self):
        new_matrix = SparseMatrix()
        # hier kommt Ihr Code
        return #...
    def print(self):
```

Dünn besetzte Matrizen spielen in sehr vielen Anwendungen eine wichtige Rolle: Beispielsweise kann man Zugverbindungen oder Straßenkarten als dünn besetzte Matrizen speichern. Dazu erstellt man eine Matrix, in der für jeden Ort eine Spalte und eine Zeile angelegt wird. Dann werden die Werte der Spalten und Zeileneinträge auf die Länge der Straßen zwischen den Orten gesetzt. (Das ergibt die sogenannte Adjazenzmatrix)

Mit Hilfe solcher Straßenmatrizen lassen sich dann unter Verwendung spezieller Algorithmen die kürzesten Wege zwischen zwei beliebigen Orten berechnen. Dazu aber mehr im nächsten Semester.

## Aufgabe 3 (Queue, *2 Punkte*)

In der Vorlesung wurde erläutert, wie man eine Queue mittels Array (mit fester Größe) implementiert. Der wesentliche Trick dabei war ja der zyklische Zugriff. Dabei kann der Fall eintreten, dass der Front- und der Back-Zeiger auf dasselbe Element zeigen.

Erklären Sie, wie man die Fälle unterscheiden kann, ob die Queue leer ist oder voll.

## Aufgabe 4 (Postfix, *2 Punkte*)

In der Vorlesung wurde der Algorithmus zum Auswerten von Postfix-Ausdrücken besprochen. Fügen Sie für den Divisions-Operator den entsprechenden Code ein.

Sie können die Aufgabe in Papierform abgeben.

## Aufgabe 5 (Stack, *2 Punkte*)

Ein Buchstabe in der nachfolgenden Sequenz steht für die Stack-Operation `push` und ein ◊ steht für die Operation `pop` mit Ausgabe des ge-pop-ten Zeichens:

E A S ◊ Y ◊ Q U E ◊ ◊ ◊ S T ◊ ◊ ◊ I O ◊ N ◊ ◊ ◊

Geben Sie die Folge der Buchstaben an, welche durch die `pop's` ausgegeben wird. Gehen Sie davon aus, dass der Stack zu Beginn leer ist.