Technische Universität Berlin **SS99**
Fachbereich Mathematik
Sekr. MA 6-1 / Frau Marcus

Dr. Michael Joswig

Olaf Jahn
Matthias Müller, André Opitz

1. Übung zur Computerorientierten Mathematik II

Abgabe: 29.04.1999 (vor der Übung)

0. Aufgabe (0/−5 Punkte)
Legt in Eurem Gruppenaccount im Unix-Pool eine Datei `~/Members` an, die die Daten Eurer Gruppenmitglieder in folgendem Format enthält (ja, in *genau* diesem Format, ohne Vertauschen von Nach- und Vorname, ohne Leerzeilen irgendwo, ohne GROSSSCHREIBEN aller Buchstaben und ohne Eintippen sonstiger Zeilen). Für jedes Gruppenmitglied steht eine Zeile der Form

`Nachname,␣Vorname,␣Matrikelnummer,␣Studiengangabkürzung`

Beispiel für eine komplette Datei:

```
Winslet, Kate, 654321, ITM
di Caprio, Leonardo, 123456, TWM
```

Fehlt die Datei am Abgabetag: −5 Punkte.

1. Aufgabe (1 Punkt)
Gebt an, wie viele Blätter ein k-ärer Baum mit n Knoten besitzt.

Ein Blatt ist ein Knoten mit Grad eins. Ein k-ärer Baum ist ein voller Wurzelbaum, in dem jeder Knoten bis auf die Blätter (die sich alle in gleicher Tiefe befinden) genau k Kinder hat.

2. Aufgabe (5 Punkte)
Entwerft eine Java-Klasse (plus evtl. Hilfsklassen) für eine doppelt verkettete Liste (von `Objects`). Sie soll natürlich alle internen Daten vor der Außenwelt verbergen, dafür aber Methoden für folgende Operationen zur Verfügung stellen:

- Konstruktion einer leeren Liste,
- Test auf Leerheit,
- Einfügen eines Elements an Anfang bzw. am Ende der Liste,
- Einfügen einer anderen Liste am Anfang bzw. Ende dieser Liste,
- Durchlauf der Liste vorwärts und rückwärts, Zugriff auf das aktuelle Element,
- Einfügen eines Element vor bzw. nach dem aktuellen Element eines Durchlaufs,
- Löschen des aktuellen Elements (was geschieht dann mit dem „Zeiger“ auf das aktuelle Element?)

Alle Methoden sollen in $O(1)$ Zeit ausgeführt werden.

Dokumentiert Eure Klasse so, wie wir es auch von Euren Programmen erwarten (siehe Programmierregeln). Beschreibt, welche Methoden in welcher Reihenfolge der Benutzer aufrufen muß, wenn er die Liste durchlaufen will. Fangt *sämtliche* Fehler ab, die ein Benutzer Eurer Klasse machen kann, und werft geeignete Exceptions aus.

3. Aufgabe (3 Punkte)

Entwickelt eine Java-Methode, die rekursiv berechnet, auf wieviele Möglichkeiten sich eine ganze Zahl s als Summe der Augenzahlen von n unterscheidbaren Würfeln darstellen läßt, wobei jeder Würfel die Augenzahlen $\{1, 2, \ldots, k\}$ aufweist.

Die Anordnung der Würfel ist wichtig, so daß es z. B. zwei Möglichkeiten für $s = 3$, $n = 2$ und $k = 2$ gibt, nämlich $(1, 2)$ und $(2, 1)$.

4. Aufgabe (5 Punkte)

Entwickelt eine Java-Methode

```
public void placeQueens(int n)
```

die das n-Damenproblem löst und die lösenden Positionen der n Damen in geeigneter Form per `System.out.println` ausgibt.

Das n-Damenproblem besteht darin, auf einem $n \times n$-Schachfeld n Damen so zu positionieren, daß keine Dame eine andere schlagen kann.

Eurer Algorithmus soll rekursiv vorgehen, wozu Ihr noch mind. eine weitere Methode (die rekursive nämlich) schreiben müßt. Die Benutzung *eines einzigen* Arrays ist erlaubt, die von Matrizen gar nicht. Gebt eine Abschätzung der Laufzeit Eures Algorithmus.

Tip: Geht spaltenweise vor und speichert im Array die Zeilenindizes der bisher gesetzten Damen. Schreibt eine Methode für den Test auf Bedrohung der neuen Dame durch die bereits gesetzten.

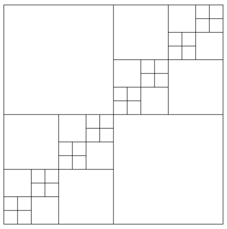
---

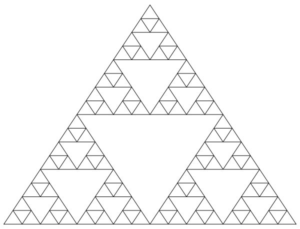
1. Programmieraufgabe

Abnahme: spätestens 29.04.99 zur Rechnervorrangzeit

Ihr sollt zwei komplett getrennte Java-Applications erstellen, wobei jedes Gruppenmitglied von Euch jeweils eine schreibt und selbst vorführt.

Die eine Application soll das linke Bild, die andere das rechte malen, wobei der Benutzer die Rekursionstiefe über ein `TextField` eingeben kann (die Abbildungen hier sind für Rekursionstiefe fünf).





Beide Programme müssen *rekursiv* vorgehen. Es dürfen keine Linien(stücke) mehrfach gemalt werden. Das Mehrfachmalen einzelner Punkte ist erlaubt.

Beim Verändern der Fenstergröße mit der Maus müssen sich die Grafiken anpassen.