Universität Mannheim
Lehrstuhl für Praktische Informatik IV
Prof. Dr. W. Effelsberg
Christoph Kuhmünch, Gerald Kühne

Praktische Informatik I
WS 1999/2000

Übungsblatt 8 **Ausgabe: Mi, 15.12.99** **Abgabe: Di, 21.12.99, 18 Uhr**

Aufgabe 1: Rekursive Algorithmen in Java [14 Punkte]

Entwickeln Sie einen Algorithmus, der einen Weg aus einem Labyrinth sucht, und implementieren Sie den Algorithmus in einem Java-Programm (unter Umständen gibt es mehrere Wege, es reicht aber, eine Lösung anzugeben). Nachfolgend ist ein Beispiel-Labyrinth abgedruckt:

```
20
20
* ******************
* * *******    *****
* *       *  *  *  *
*   ***** * **     *
***   *     **  ** *
******* * *******  *
*     ***** *   *  *
* * ****    ** ** **
***           ** **
*   **+ * *    ** **
*  **   * *** *** **
**  * *** *   *** **
*   ***** *    ** **
*     **  ******   *
* ***      *  *  ***
*  **** *     * ****
*     * * ***   ****
*  *  *** * * ******
*  * **     *      *
******************
```

Sterne (“*”) stellen eine Mauer dar, Wege sind durch Leerzeichen (“ ”) gekennzeichnet. Das Plus-Zeichen(“+”) symbolisiert den Standpunkt im Labyrinth zum Startzeitpunkt. Gehen Sie davon aus, daß lediglich Bewegungen in horizontaler und vertikaler Richtung möglich sind, diagonal Bewegungen sind nicht möglich.

Anmerkung: Die Abgabe des Programmes muß per Abox erfolgen! Benutzen Sie den Klassennamen `Labyrinth`! Das Programm muß als Kommandozeilenparameter den Namen einer Datei erwarten, die ein Labyrinth in der oben abgebildeten Form enthält. Ein Beispiel für eine solche Kommandodatei finden Sie im Web[1]. Die ersten beiden Zeilen enthalten die Höhe und Breite des Labyrinths. Als Ergebnis muß ihr Programm folgende Ausgaben liefern:

Wenn ein Weg zum Ausgang gefunden wurde, muß in der ersten Zeile “Ausgang gefunden!” ausgegeben werden. In den nachfolgenden beiden Zeilen muß die Position des Ausgangs angegeben werden (zuerst die Zeile, dann die Spalte). Ursprung des Koordinatensystems (0,0) ist die linke obere Ecke des Labyrinthes. In den nachfolgenden Zeilen muß das Labyrinth mit

[1] http://www.informatik.uni-mannheim.de/informatik/pi4/stud/veranstaltungen/ws199900/pi1/ueb/blatt8/labyrinth.txt

eingezeichnetem Lösungsweg ausgeben werden. Benutzen Sie zur Markierung die Ziffer Null (“0”).

Falls kein Ausgang gefunden wurde, muß das Programm “Es gibt keinen Weg zum Ausgang!” ausgeben.

Hier ein Beispiel für einen Kommandozeilenaufruf und der Ergebnisausgabe des von Ihnen zu entwickelnden Programms:

```
 > java Labyrinth labyrinth.txt
Ausgang gefunden!
0 Zeile
1 Spalte
*0******************
*0* ******* 000*****
*0*0000000*00*00*00*
*000*****0*0** 0000*
***   *  000**  **0*
******* * *******00*
*     ***** *   *0 *
* * ****    ** **0**
***0000        **0**
* 00**0 * *    **0**
* 0**   * *** ***0**
**0 * *** *   ***0**
* 0 ***** *    **0**
* 0000**00******00 *
* ***00000 *00*00***
*  **** *00000*0****
*     * * ***000****
*  *  *** * * ******
*  * **     *      *
********************
```

Aufgabe 2: Fibonacci-Zahlen [6 Punkte]

In Übungsblatt 7 wurde die Fibonacci-Funktion wie folgt definiert:

$$fib(n) = \begin{cases} 1 & , \quad \textit{falls } n = 1 \\ 1 & , \quad \textit{falls } n = 2 \\ fib(n-1) + fib(n-2) & , \quad \textit{sonst} \end{cases} \tag{1}$$

Der nachfolgende Algorithmus ist aus der Musterlösung von Blatt 7 entnommen (Fehler-Abfrage wurde absichtlich ausgelassen):

```
long fibo (long n) {
  if (n <= 2) {
    return 1;
  } else {
    return fibo (n-1) + fibo (n-2);
  }
}
```

(a) [3 Punkte] Zeigen Sie, daß der oben abgebildete Algorithmus terminiert!

(b) [3 Punkte] Beweisen Sie, daß der oben abgebildete Algorithmus korrekt ist!