TECHNISCHE UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR INFORMATIK

**Lehrstuhl für Sprachen und Beschreibungsstrukturen**
**Praktikum Grundlagen der Programmierung**
Prof. Dr. Helmut Seidl

WS 2007/2008
Aufgabenblatt 13
25. Januar 2008

**Abgabe:** Mo, 18.02.2008 zentrale Abgabe über https://grundstudium.in.tum.de/info1abgabe

# Praktikum Grundlagen der Programmierung

## Aufgabe 58 (Ü) Fotoalbum

Entwickeln Sie ein Applet, das eine Folge von mindestens drei Bildern darstellt. Das Applet soll jeweils ein Bild zusammen mit dem Namen der Datei, in der sich das Bild befindet, darstellen. Außerdem soll es einen Knopf zur Verfügung stellen, durch den das nächste Bild per Mausklick angezeigt werden kann.

## Aufgabe 59 (Ü) Tabellenkalkulation

In dieser Aufgabe sollen Sie ein vereinfachtes Modell einer Tabellenkalkulation in Java entwerfen. Grundelement der Tabellenkalkulation ist eine Tabelle, die aus Zellen besteht — organisiert in 30 Zeilen und 30 Spalten. In jeder Zelle kann ein Ausdruck stehen. Ein Ausdruck kann dabei folgendes sein:

- eine `int` -Konstante (Klasse `Const`), z.B. 5;
- Die Summe zweier Ausdrücke (Klasse `Add`);
- das additive Inverse eines Teilausdrucks (Klasse `UnMinus`) oder
- ein Verweis auf eine andere Zelle (Klasse `Ref`) der Tabelle.

a) Implementieren Sie die geschilderte Klassenhierarchie zur Repräsentation von Ausdrücken. Verwenden Sie eine gemeinsame Basisklasse `Expr`, die eine abstrakte Methode `public abstract int eval()` zur Auswertung des Ausdrucks deklariert.

   Gehen Sie zur Implementation von `Ref` davon aus, dass die Klasse `Tabelle` die in b) beschriebene Methode `int evalZelle(int i, int j)` zu Verfügung stellt.

b) Implementieren Sie die Klasse `Tabelle`. Auf den Inhalt der Tabelle darf nur über die folgenden Methoden zugegriffen werden:

   - `int evalZelle(int i, int j)`
     Der Aufruf `evalZelle(i,j)` gibt den Wert des in Zelle $(i, j)$ gespeicherten Ausdrucks zurück. Um doppelte Berechnungen zu vermeiden, soll ein bereits berechnetes Ergebnis für jede Zelle gespeichert werden. Nachfolgende Aufrufe von `evalZelle()` liefern das gespeicherte Ergebnis zurück.
   - `void setExpression(int i, int j, Expr e)`
     Der Aufruf `setExpression(i,j,e)` speichert den Ausdruck `e` in Zelle $(i, j)$ ab. Damit werden alle gespeicherten Ergebnisse potentiell ungültig und müssen daher zurückgesetzt werden.

## Aufgabe 60 (H) Bombensuche (15+5 Punkte)

In dieser Aufgabe sollen Sie das Spiel Bombensuche entwickeln, in dem Sie auf einem Spielbrett mit verdeckten Feldern alle Bomben möglichst schnell finden müssen. Hinter jedem aufzudeckenden Feld verbirgt sich entweder eine Bombe oder eine Zahl. Die Zahl gibt die Anzahl an Bomben an, die sich in den angrenzenden 8 Feldern des aufzudeckenden Feldes befinden. Ziel des Spieles ist es, alle Felder unter denen sich keine Bombe befindet, aufzudecken. Ein Feld kann aufgedeckt werden, indem der Spieler einen einfachen Mausklick auf das Feld tätigt. Wenn das aufgedeckte Feld verbombt ist, dann hat der Spieler das Spiel verloren. Möchte der Spieler ein Feld markieren, von dem er annimmt, dass sie eine Bombe enthält, so muss er mit der rechten Maustaste einen Klick auf dieses Feld ausführen.

Zudem enthält der Spielbereich eine Uhr und einen Punktezähler. Für jedes korrekt aufgedeckte Feld werden Pluspunkte auf den Punktezähler addiert. Vor Ablauf der Uhr müssen alle nicht verbombten Felder aufgedeckt bzw. alle verbombten Felder markiert sein.

Wie die Punktezählung im Detail aussieht, bleibt Ihnen überlassen.

Schreiben Sie zur Lösung dieser Aufgabe ein Applet, welches das Spiel Bombensuche simuliert. Hierbei soll der Spieler die Grösse seines Spielfeldes, die Anzahl der enthaltenen Bomben und die maximale Spielzeit zum Aufdecken der Felder angeben können.

**Zusatz:**
Nach etwa dreiviertel der Spielzeit werden die Bomben scharf, d.h. die Bomben hinter jedem nicht vom Spieler als verbombt klassifizierten Feld explodieren und zwar in einem zeitlichen Abstand von etwa $2-5$ Sekunden. Die Explosion bringt Minuspunkte für den Punktestand des Spielers. Zudem hat die Explosion zur Folge, dass einige der 8 angrenzenden Felder des Bombenfeldes verschüttet werden, d.h. diese werden wieder umgedreht.