Hausaufgabe 3 (1) (6 Punkte)

Abgabe: 17. Juni 2012

Entwickeln Sie einen **ADT** *Matrix*, dessen Instanzen **beliebige** m×n-Matrizen **ganzer** Zahlen darstellen können! Übergeben Sie die Dimensionen der Matrix im Konstruktor! Implementieren Sie die Funktionalität der im folgenden gegebenen Schnittstelle!

- `set(int* a)`: Einlesen der Werte der Matrix aus dem eindimensionalen Feld a
- `out()`: Ausgabe der Werte der eigenen Instanz als Matrix formatiert auf dem Bildschirm
- `trp()`: Transponierung der eigenen Instanz
- `add(Matrix& m)`: Matritzenaddition von m zur eigenen Instanz
- `mul(Matrix& m)`: Matritzenmultiplikation der eigenen Instanz mit der Matrix m und Ausgabe des Ergebnisses auf dem Bildschirm (Das Ergebnis muss nicht gespeichert werden)

Geben Sie für trp(), add() und mul() die Komplexitätsklasse bzgl. der Anzahl der Elemente der Matrix an!

Zusatzaufgabe

- `sym()`: Test, ob die eigene Instanz symmetrisch ist
- `ort()`: Test, ob die eigene Instanz orthogonal ist
- Geben Sie für ort() und sym() die Komplexitätsklasse bzgl. der Anzahl der Elemente der Matrix an!

Hausaufgabe 3 (2) (6 Punkte)

Abgabe: 17. Juni 2012

Anwendung des ADTs

Schreiben Sie ein Programm, das aus zwei via Kommandozeile übergebenen Textdateien die Dimensionen und Werte zweier Matrizen einliest. Erzeugen Sie zwei Instanzen Ihres ADTs und initialisieren Sie diese mit den eingelesenen Werten mit Hilfe der set()-Methode.

Führen Sie nacheinander folgende Aktionen auf Ihren ADTs aus:

- Aufruf der out()-Methode der ersten Instanz und danach Aufruf der out()-Methode der zweiten Instanz

- Addition der zweiten Matrix zur ersten Matrix

- Aufruf der out()-Methode der ersten Matrix

- Multiplikation der ersten mit der zweiten Matrix

- Transponierung der ersten Matrix

- Aufruf der out()-Methode der ersten Matrix

Zusatzaufgabe

- Test auf Symmetrie der zweiten Matrix mit entsprechender Bildschirmausgabe

- Test auf Orthogonalität der Matrix mit entsprechender Bildschirmausgabe

Hausaufgabe 3 (3) (6 Punkte)

Abgabe: 17. Juni 2012

Beachten Sie:

- die Aufgabe ist **ausschließlich als ADT** zu realisieren
- Bibliotheksfunktionen sind **ausschließlich für Ein- und Ausgabe** erlaubt
- Ihr Quelltext muss sich mit **GCC** kompilieren lassen
- Fehler beim Einlesen aus der Datei, z.B. aufgrund eines falschen Dateiformats müssen Sie nicht abfangen, ebenso können Sie davon ausgehen, dass genügend Speicher für Ihre Matrizen zur Verfügung steht
- Um die formatierte Darstellung einer Matrix zu erleichtern, können Sie die davon ausgehen, dass keine Werte > 999 oder < -999 dargestellt werden müssen
- die Textdatei ist wie folgt aufgebaut:
    - zuerst wird die Anzahl der Zeilen der Matrix spezifiziert, danach die Anzahl der Spalten
    - danach folgen die Werte der einzelnen Matrixelemente
        - beginnend mit den Elementen der ersten Zeile,
        - danach den Elementen der zweiten Zeile usw.
    - einzelne Werte können durch Leerzeichen oder Zeilenvorschub voneinander getrennt sein

Hausaufgabe 3 (4) (6 Punkte)

Abgabe: 17. Juni 2012

Beispiel 1

Dateiname: `Matrix.txt`

Dateiinhalt: `3 2 6 -1 3 2 0 -3`

Im Hauptprogramm wird eine Instanz des ADT Matrix mit 3 Zeilen und 2 Spalten erzeugt, der Matrixinhalt gelesen und die Werte über die set()-Methode gesetzt. Die formatierte Ausgabe dieser Matrix mittels der out()-Methode soll folgendes Ergebnis liefern:

```
 6   -1
 3    2
 0   -3
```

eine Transponierung dieser Matrix mittel trp()-Methode und anschließende Ausgabe mittels out() liefern

```
 6    3    0
-1    2   -3
```

Hausaufgabe 3 (5) (6 Punkte)

Abgabe: 17. Juni 2012

Beispiel 2

Dateiname: `AndereMatrix.txt`

Dateiinhalt: `2 3 1 2 3 4 5 6`

Im Hauptprogramm wird eine Instanz des ADT Matrix mit 2 Zeilen und 3 Spalten erzeugt, der Matrixinhalt gelesen und die Werte über die set()-Methode gesetzt. Die formatierte Ausgabe dieser Matrix mittels der out()-Methode soll folgendes Ergebnis liefern:

```
1    2    3
4    5    6
```

Der Aufruf der mul()-Methode der Matrix aus Beispiel 1 mit dem Argument der Matrix aus Beispiel 2 soll folgende Bildschirmausgabe liefern:

```
12   -6
39  -12
```