

Prüfungsvorleistung 4 (6+2 Punkte)

Abgabe: 28. Juni 2015

Entwickeln Sie einen ADT Matrix, dessen Instanzen beliebige $M \times N$ -Matrizen ganzer Zahlen darstellen können!

Übergeben Sie die Zeilen- und Spaltenanzahl der Matrix im Konstruktor und 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 der Matrix `m` zur eigenen Instanz, d.h. die eigene Instanz beinhaltet im Anschluss das Ergebnis der Addition
- `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 Matrixelemente 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!

Prüfungsvorleistung 4 (6+2 Punkte)

Abgabe: 28. Juni 2015

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 von Matrix 1
- Aufruf der `out ()`-Methode von Matrix 2
- Addition von Matrix 1 mit Matrix 2 (Funktionsargument von `add ()` ist Matrix 2)
- Aufruf der `out ()`-Methode von Matrix 1
- Multiplikation von Matrix 2 mit Matrix 1 (Funktionsargument von `mul ()` ist Matrix 1)
- Transponierung von Matrix 1
- Aufruf der `out ()`-Methode von Matrix 1

Zusatzaufgabe

- Teste Matrix 2 auf Symmetrie und gib das Ergebnis auf dem Bildschirm aus.
- Teste Matrix 1 auf Orthogonalität und gib das Ergebnis auf dem Bildschirm aus.

Prüfungsvorleistung 4 (6+2 Punkte)

Abgabe: 28. Juni 2015

Beachten Sie:

- Die Matrix ist ausschließlich als ADT zu realisieren!
- Bibliotheksfunktionen sind ausschließlich für Ein- und Ausgabe erlaubt!
- Fehler beim Einlesen aus der Datei, z.B. aufgrund eines falschen Dateiformats müssen Sie nicht abfangen.
- Sie können davon ausgehen, dass genügend Speicher für Ihre Matrizen zur Verfügung steht.
- Für Matrixoperationen, die z.B. aufgrund inkompatibler Dimensionierung nicht möglich sind, geben Sie eine entsprechende aussagekräftige Fehlermeldung aus!
- Um die formatierte Darstellung einer Matrix zu erleichtern, können Sie 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

Prüfungsvorleistung 4 (6+2 Punkte)

Abgabe: 28. Juni 2015

Beispiel für Matrix 1

Dateiname: `Matrix1.txt`

Dateiinhalte: 3 2 6 -1 3 2 0 -3

Im Hauptprogramm wird eine Instanz des ADTs *Matrix* mit 3 Zeilen und 2 Spalten erzeugt, der Matrixinhalt gelesen und die Werte über die `set ()`-Methode gesetzt.

Die formatierte Ausgabe von Matrix 1 mittels der `out ()`-Methode sollte folgendes Ergebnis liefern:

6	-1
3	2
0	-3

Eine Transponierung von Matrix 1 mittels `trp ()`-Methode und anschließende Ausgabe mittels `out ()` sollte folgendes Ergebnis liefern:

6	3	0
-1	2	-3

Prüfungsvorleistung 4 (6+2 Punkte)

Abgabe: 28. Juni 2015

Beispiel für Matrix 2

Dateiname: `Matrix2.txt`

Dateiinhalte: 2 3 1 2 3 4 5 6

Im Hauptprogramm wird eine Instanz des ADTs *Matrix* mit 2 Zeilen und 3 Spalten erzeugt, der Matrixinhalt gelesen und die Werte über die `set ()`-Methode gesetzt.

Die formatierte Ausgabe von Matrix 2 mittels der `out ()`-Methode sollte folgendes Ergebnis liefern:

1	2	3
4	5	6

Der Aufruf der `mul ()`-Methode von Matrix 2 mit Matrix 1 sollte folgende Bildschirmausgabe liefern:

12	-6
39	-12