TU Berlin, Institut für Mathematik
Thomas Slawig, Uwe Prüfert, Henrik Büsing

10. Übungsblatt

Abgabe Theorie bis zum 24.1.06

`www.math.tu-berlin.de/Vorlesungen/WS05/ProgMa`

1. Aufgabe (8 Punkte)

Zeichnen Sie ein UML-Diagramm für eine Klasse `Matrix`, die reelle Matrizen beliebiger Dimensionen $(m \times n)$ mit $m, n \in \mathbb{N} \setminus \{0\}$ repräsentieren soll. Es sollen zur Verfügung stehen: Je eine Methode zur

- Eingabe einer Matrix über die Tastatur,
- Ausgabe einer Matrix auf den Bildschirm,
- Ausgabe von Zeilen- und Spaltenanzahl,
- Hinzufügen einer Zeile an beliebiger Stelle,
- Hinzufügen einer Spalte an beliebiger Stelle,
- Streichen einer Zeile an beliebiger Stelle,
- Streichen einer Spalte an beliebiger Stelle,
- Lösen des linearen Gleichungssystems $AX = B$, wobei A, X, B als Matrizen interpretiert werden sollen,

und die Operatoren:

- Addition, Subtraktion, Multiplikation,
- unärer Negationsoperator,
- Vergleichsoperator ==.

Das Diagramm soll die Attribute, eventuell vorhandene Zusicherungen sowie Namen, evtl. Parameter und Rückgabewerte mit Typen der Methoden und Operatoren darstellen. Konstruktoren und Destruktoren müssen nicht aufgeführt werden.

2. Aufgabe (8 Punkte)

Leiten Sie von der in Aufgabe 1 definierten Klasse eine Unterklasse ab, mit der quadratische Matrizen sinnvoll beschrieben werden können.

- Beachten Sie dabei die folgende Regel: **Es soll keine einschränkende Vererbung durchgeführt werden, d.h. die Unterklasse soll keine Zusicherungen auf Attribute der Oberklasse enthalten.**
- Welche Methoden der Oberklasse müssen überschrieben werden?
- Es sollen folgende zusätzliche Methoden zur Verfügung stehen:
  - Berechnung der Determinante.
  - Berechnung der Spur der Matrix (spur $A := \sum_i a_{ii} \in \mathbb{R}$).
- Zeichnen Sie ein UML-Diagramm der Unterklasse. Es soll die Attribute, eventuell vorhandene Zusicherungen sowie Namen, evtl. Parameter und Rückgabewerte mit Typen der Methoden und Operatoren darstellen. Konstruktoren und Destruktoren müssen nicht aufgeführt werden.

8. Programmieraufgabe (Vorführen bis zum 24.1.06)

Ergänzen Sie die Klasse `Bruch` um die Operatoren − (unär und binär), /, += und == sowie um eine Methode, die einen Bruch mit einer ganzen Zahl erweitert.