template Matrixklasse

Einführung:

Diese Klasse repräsentiert eine Matrix und ermöglicht verschiedene Operationen an Matrices durchzuführen. (Die Benützer soll um die Speicherallokation und Freigabe oder um die Operation der Funktionen in der Klasse nicht sorgen.)

Zur Instanziierung eines Objekts. Die Benützer:

1. soll die Typ der Elemente des Matrices bestimmen. Auf diese Typ soll Multiplikation und Addition definiert werden, um die Funktionalität der Klasse zu sichern.
2. soll die Anzahl der Zeilen und Spalten bestimmen.
3. kann die Initialisierungswert bestimmen, falls es nicht bestimmt wird, ist alle Elemente auf 0 initialisiert.

Syntax: MyMatrix<Typ> Name(Zeilen, Spalten, Initialisierungswert);

Verwendung des Objekts:

Definierte Operationen:

Operationen, die den Zustand der Instanz nicht ändert (Ergebnisse: nicht gespeichert)

- Kopie eine Matrix herstellen
  - Syntax: MyMatrix<Typ> NeuerName (Name), wo Name ist der Name eine schon existierende Instanz.
- Indexierung mit Hilfe von dem Operator ()
  - Syntax: Name (Zeilenzahl, Spaltenzahl)
- rows(), Gibt die Anzahl der Zeilen zurück.
  - Syntax: Name.rows();
- columns(), gibt die Anzahl der Spalten zurück.
  - Syntax: Name.columns();
- Matrix + Matrix, wo die Typen und Größe der Matrices sind die gleichen.
  - Syntax: Name + Nam2;
- Matrix * Skalar, wo der Typ der Skalar soll mit dem Typ der Elemente der Matrices gleich sein.
  - Syntax: Name * 5;
- Skalar * Matrix, wo der Typ der Skalar soll mit dem Typ der Elemente der Matrices gleich sein.
  - Syntax: 5 * Name;
- Matrix *= Matrix, wo die Anzahl der Spalten der ersten Matrix soll mit der Anzahl der Zeilen der zweiten Matrix gleich sein. ○ Syntax: Name * Name2;
- transposition(), gibt die transponierte Matrix zurück.
  - Syntax: Name.transposition();
- rank (), gibt die Rank der Matrix zurück.
  - Syntax: Name.rank();
- determinant(), gibt die Determinant der Matrix zurück.
  - Syntax: Name.determinant();

Operationen, die den Zustand der Instanz ändert (Ergebnisse: gespeichert):

- Matrix += Matrix, wo die Typen und Größe der Matrices sind die gleichen.
    - Syntax: Name += Name2;
- Matrix *= Skalar, wo der Typ der Skalar soll mit dem Typ der Elemente der Matrices gleich sein.
    - Syntax: Name *= 2;
- Matrix *= Matrix, wo die Anzahl der Spalten der ersten Matrix soll mit der Anzahl der Zeilen der zweiten Matrix gleich sein. ○ Syntax: Name *= Name2
- Matrix = Matrix, die Typ der zwei Matrices soll gleich sein.
    - Syntax: Name = Name2;
- rowAdd(row_dest, row_source), addiert die Zeile row_source to die Zeile row_dest.
    - Syntax: Name.rowAdd(0,1);
- rowInterchange(row1, row2), vertauscht die Zeile row1 mit der Zeile row2. ○ Syntax: Name.rowInterchange(1,2);
- rowMult(row_dest, value), multipliziert alle Elemente der Zeile row_dest mit der Wert value.
    - Syntax: Name.rowMult(0,134);