

TemplateMatrixklasse

Generated by Doxygen 1.8.18

1 Namespace Index	1
1.1 Namespace List	1
2 Class Index	3
2.1 Class List	3
3 Namespace Documentation	5
3.1 GHA Namespace Reference	5
3.1.1 Detailed Description	5
4 Class Documentation	7
4.1 GHA::MyMatrix< T > Class Template Reference	7
4.1.1 Constructor & Destructor Documentation	8
4.1.1.1 MyMatrix() [1/2]	8
4.1.1.2 MyMatrix() [2/2]	8
4.1.1.3 ~MyMatrix()	8
4.1.2 Member Function Documentation	8
4.1.2.1 calcdet()	9
4.1.2.2 columns()	9
4.1.2.3 determinant()	9
4.1.2.4 operator()() [1/2]	9
4.1.2.5 operator()() [2/2]	10
4.1.2.6 operator*() [1/2]	10
4.1.2.7 operator*() [2/2]	10
4.1.2.8 operator*=() [1/2]	11
4.1.2.9 operator*=() [2/2]	11
4.1.2.10 operator+()	12
4.1.2.11 operator+=()	12
4.1.2.12 operator=()	12
4.1.2.13 rank()	13
4.1.2.14 rowAdd()	13
4.1.2.15 rowInterchange()	13
4.1.2.16 rowMult()	14
4.1.2.17 rows()	14
4.1.2.18 transposition()	14
4.1.3 Member Data Documentation	15
4.1.3.1 m_column	15
4.1.3.2 m_row	15
4.1.3.3 pMatrix	15
Index	17

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

GHA	5
-------------------------------	---

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

GHA::MyMatrix< T >	7
--	---

Chapter 3

Namespace Documentation

3.1 GHA Namespace Reference

Classes

- class [MyMatrix](#)

3.1.1 Detailed Description

Die Klasse [MyMatrix](#) repräsentiert ein Matrix und ermöglicht verschiedene Operationen an Matrizen durch zu führen.

Diese Matrix innerhalb der Klasse wird als ein eindimensionale, zusammenhängendes Array dynamisch gespeichert, und bei der Löschen der Instanz wird es freigegeben.

Bei theoretischen Fehlern gibt die Klasse eine Ausnahme.

Chapter 4

Class Documentation

4.1 GHA::MyMatrix< T > Class Template Reference

Public Member Functions

- [MyMatrix](#) (unsigned int row, unsigned int column, T value=0)
- [MyMatrix](#) (const [MyMatrix](#)< T > &other)
- [~MyMatrix](#) ()
- T & [operator\(\)](#) (unsigned int row, unsigned int column)
- T [operator\(\)](#) (unsigned int row, unsigned int column) const
- unsigned int [rows](#) () const
- unsigned int [columns](#) () const
- [MyMatrix](#)< T > & [operator=](#) (const [MyMatrix](#)< T > &other)
- [MyMatrix](#)< T > [operator+](#) (const [MyMatrix](#)< T > &other) const
- [MyMatrix](#)< T > & [operator+=](#) (const [MyMatrix](#)< T > &other)
- [MyMatrix](#)< T > [operator*](#) (T other) const
- [MyMatrix](#)< T > & [operator*=](#) (T other)
- [MyMatrix](#)< T > [operator*](#) (const [MyMatrix](#)< T > &other) const
- [MyMatrix](#)< T > & [operator*=](#) (const [MyMatrix](#)< T > &other)
- [MyMatrix](#)< T > & [rowAdd](#) (unsigned int row_dest, unsigned int row_source)
- [MyMatrix](#)< T > & [rowMult](#) (unsigned int row_dest, T value)
- [MyMatrix](#)< T > & [rowInterchange](#) (unsigned int row1, unsigned int row2)
- [MyMatrix](#)< T > [transposition](#) () const
- unsigned int [rank](#) ()
- T [determinant](#) ()

Private Member Functions

- T [calcdet](#) ([MyMatrix](#)< T > matrix, unsigned int n)

Private Attributes

- unsigned int [m_row](#)
- unsigned int [m_column](#)
- T * [pMatrix](#)

4.1.1 Constructor & Destructor Documentation

4.1.1.1 MyMatrix() [1/2]

```
template<typename T >
GHA::MyMatrix< T >::MyMatrix (
    unsigned int row,
    unsigned int column,
    T value = 0 )
```

Die Konstrukter der Klasse, vorbereitet die Matrix auf Verwendung.

Parameters

in	<i>row</i>	Die Anzhal der Zeilen innerhalb der Klasse
in	<i>column</i>	Die Anzahl der Spalten innerhalb der Klasse
in	<i>value</i>	Alle Elementen des Matrices weden auf dieser Wert inicializiert, standardmäßig: 0.

4.1.1.2 MyMatrix() [2/2]

```
template<typename T >
GHA::MyMatrix< T >::MyMatrix (
    const MyMatrix< T > & other )
```

Die Copykonstruktor der Klasse, nimmt ein schon existierende Matrix und kopiert es.

Parameters

in	<i>other</i>	Ein schon existierende Matrix, das wir kopieren möchten.
----	--------------	--

4.1.1.3 ~MyMatrix()

```
template<typename T >
GHA::MyMatrix< T >::~~MyMatrix ( ) [inline]
```

Die Destruktor der Klasse, löscht die dynamisch allokierte Speicherplatz.

4.1.2 Member Function Documentation

4.1.2.1 calcdet()

```
template<typename T >
T GHA::MyMatrix< T >::calcdet (
    MyMatrix< T > matrix,
    unsigned int n ) [private]
```

Hilfsfunktion für die rekursive Berechnung der Determinante.

4.1.2.2 columns()

```
template<typename T >
unsigned int GHA::MyMatrix< T >::columns ( ) const [inline]
```

Gibt die Anzahl der Spalten innerhalb die Matrix zurück.

Returns

Anzahl der Spalten

4.1.2.3 determinant()

```
template<typename T >
T GHA::MyMatrix< T >::determinant
```

Gibt die Determinante des Matrix zurück.

Returns

Determinante des Matrix

4.1.2.4 operator>() [1/2]

```
template<typename T >
T & GHA::MyMatrix< T >::operator() (
    unsigned int row,
    unsigned int column )
```

Indexierung und Zugriff des Matrices wird durch die () Operator verwirklicht.

Parameters

in	<i>row</i>	Die Zeilenzahl
in	<i>column</i>	Die Spaltenzahl

Returns

Referenz auf das gegebenen Element

4.1.2.5 operator() [2/2]

```
template<typename T >
T GHA::MyMatrix< T >::operator() (
    unsigned int row,
    unsigned int column ) const
```

Indexierung des Matrices wird durch die () Operator verwirklicht.

Parameters

in	<i>row</i>	Die Zeilenzahl
in	<i>column</i>	Die Spaltenzahl

Returns

Wert des gegebenen Elements

4.1.2.6 operator*() [1/2]

```
template<typename T >
MyMatrix< T > GHA::MyMatrix< T >::operator* (
    const MyMatrix< T > & other ) const
```

Matrixmultiplikation.

Parameters

in	<i>other</i>	Ein Matrix mit gleicher Typ
----	--------------	-----------------------------

Returns

Ergebnismatrix als Wert

4.1.2.7 operator*() [2/2]

```
template<typename T >
MyMatrix< T > GHA::MyMatrix< T >::operator* (
    T other ) const
```

Überladene Multiplikationsoperator

Parameters

<i>in</i>	<i>other</i>	Ein Wert, die die gleiche Typ hat als die Elementen des Matrices
-----------	--------------	--

Returns

Ergebnismatrix als Wert

4.1.2.8 operator*=() [1/2]

```
template<typename T >
MyMatrix< T > & GHA::MyMatrix< T >::operator*= (
    const MyMatrix< T > & other )
```

Matrixmultiplikation mit Zuweisung.

Parameters

<i>in</i>	<i>other</i>	Ein Matrix mit gleicher Typ
-----------	--------------	-----------------------------

Returns

Ergebnismatrix als Referenz

4.1.2.9 operator*=() [2/2]

```
template<typename T >
MyMatrix< T > & GHA::MyMatrix< T >::operator*= (
    T other )
```

Überladene Multiplikationsoperator mit Zuweisung.

Parameters

<i>in</i>	<i>other</i>	Ein Wert, die die gleiche Typ hat als die Elementen des Matrices
-----------	--------------	--

Returns

Ergebnismatrix als Referenz

4.1.2.10 operator+()

```
template<typename T >
MyMatrix< T > GHA::MyMatrix< T >::operator+ (
    const MyMatrix< T > & other ) const
```

Überladene Additionsoperator auf zwei Matrizen.

Parameters

in	<i>other</i>	Ein Matrix mit gleicher Typ
----	--------------	-----------------------------

Returns

Ergibnismatrix als Wert

4.1.2.11 operator+=()

```
template<typename T >
MyMatrix< T > & GHA::MyMatrix< T >::operator+= (
    const MyMatrix< T > & other )
```

Überladene Additionsoperator mit Zuweisung.

Parameters

in	<i>other</i>	Ein Matrix mit gleicher Typ
----	--------------	-----------------------------

Returns

Referenz auf die lvalue mit neuer Wert

4.1.2.12 operator=()

```
template<typename T >
MyMatrix< T > & GHA::MyMatrix< T >::operator= (
    const MyMatrix< T > & other )
```

Überladene Zuweisungsoperator auf zwei Matrizen.

Parameters

in	<i>other</i>	Ein Matrix mit gleicher Typ
----	--------------	-----------------------------

Returns

Referenz auf die lvalue mit neuer Wert

4.1.2.13 rank()

```
template<typename T >
unsigned int GHA::MyMatrix< T >::rank
```

Gibt die Rank des Matrix zurück.

Returns

Rank des Matrix

4.1.2.14 rowAdd()

```
template<typename T >
MyMatrix< T > & GHA::MyMatrix< T >::rowAdd (
    unsigned int row_dest,
    unsigned int row_source )
```

Addition von zwei Zeilen.

Parameters

in	<i>row_dest</i>	Zeilenindex der Zeile, zu der die andere addiert wird
in	<i>row_source</i>	Zeilenindex der Zeile, die zu der andere addiert wird

Returns

Ergebnismatrix als Referenz

4.1.2.15 rowInterchange()

```
template<typename T >
MyMatrix< T > & GHA::MyMatrix< T >::rowInterchange (
    unsigned int row1,
    unsigned int row2 )
```

Vertauschen von zwei Zeilen.

Parameters

in	<i>row1</i>	Zeilenindex einer Zeile
in	<i>row2</i>	Zeilenindex der andere Zeile

Returns

Ergebnismatrix als Referenz

4.1.2.16 rowMult()

```
template<typename T >
MyMatrix< T > & GHA::MyMatrix< T >::rowMult (
    unsigned int row_dest,
    T value )
```

Zeilenmultiplikation mit einer Wert.

Parameters

in	<i>row_dest</i>	Zeilenindex der Zeile, die mit der Wert multipliziert wird
in	<i>value</i>	Mit dieser Wert wird die Zeile multipliziert

Returns

Ergebnismatrix als Referenz

4.1.2.17 rows()

```
template<typename T >
unsigned int GHA::MyMatrix< T >::rows ( ) const [inline]
```

Gibt die Anzahl der Zeilen innerhalb die Matrix zurück.

Returns

Anzahl der Zeilen

4.1.2.18 transposition()

```
template<typename T >
MyMatrix< T > GHA::MyMatrix< T >::transposition
```

Transponierung des Matrix.

Returns

Ergebnismatrix als Wert.

4.1.3 Member Data Documentation

4.1.3.1 m_column

```
template<typename T >
unsigned int GHA::MyMatrix< T >::m_column [private]
```

Anzahl von Spalten innerhalb des Matrix.

4.1.3.2 m_row

```
template<typename T >
unsigned int GHA::MyMatrix< T >::m_row [private]
```

Anzahl von Zeilen innerhalb des Matrix.

4.1.3.3 pMatrix

```
template<typename T >
T* GHA::MyMatrix< T >::pMatrix [private]
```

Zeiger auf die Elementen des Matrix.

The documentation for this class was generated from the following file:

- C:/Users/kusti/source/repos/GHA2/GHA2/MyMatrix.h

Index

~MyMatrix
GHA::MyMatrix< T >, 8

calcdet
GHA::MyMatrix< T >, 8

columns
GHA::MyMatrix< T >, 9

determinant
GHA::MyMatrix< T >, 9

GHA, 5
GHA::MyMatrix< T >, 7
~MyMatrix, 8
calcdet, 8
columns, 9
determinant, 9
m_column, 15
m_row, 15
MyMatrix, 8
operator*, 10
operator*=: 11
operator(), 9, 10
operator+, 11
operator+=, 12
operator=, 12
pMatrix, 15
rank, 13
rowAdd, 13
rowInterchange, 13
rowMult, 14
rows, 14
transposition, 14

m_column
GHA::MyMatrix< T >, 15

m_row
GHA::MyMatrix< T >, 15

MyMatrix
GHA::MyMatrix< T >, 8

operator*
GHA::MyMatrix< T >, 10

operator*=
GHA::MyMatrix< T >, 11

operator()
GHA::MyMatrix< T >, 9, 10

operator+
GHA::MyMatrix< T >, 11

operator+=
GHA::MyMatrix< T >, 12

operator=
GHA::MyMatrix< T >, 12

pMatrix
GHA::MyMatrix< T >, 15

rank
GHA::MyMatrix< T >, 13

rowAdd
GHA::MyMatrix< T >, 13

rowInterchange
GHA::MyMatrix< T >, 13

rowMult
GHA::MyMatrix< T >, 14

rows
GHA::MyMatrix< T >, 14

transposition
GHA::MyMatrix< T >, 14