

template CMatrice

Généré par Doxygen 1.8.13

Table des matières

1	Index des classes	1
1.1	Liste des classes	1
2	Index des fichiers	3
2.1	Liste des fichiers	3
3	Documentation des classes	5
3.1	Référence de la classe CException	5
3.1.1	Documentation des constructeurs et destructeur	5
3.1.1.1	CException() [1/3]	5
3.1.1.2	CException() [2/3]	5
3.1.1.3	CException() [3/3]	6
3.1.2	Documentation des fonctions membres	6
3.1.2.1	sEXPmessageException()	6
3.1.2.2	uiEXPgetType()	6
3.2	Référence du modèle de la classe CMatrice< Mtype >	7
3.2.1	Documentation des constructeurs et destructeur	7
3.2.1.1	CMatrice() [1/3]	7
3.2.1.2	CMatrice() [2/3]	8
3.2.1.3	CMatrice() [3/3]	8
3.2.2	Documentation des fonctions membres	8
3.2.2.1	MATafficherMatrice()	8
3.2.2.2	MATlireFichier()	9
3.2.2.3	MATtransposee()	9
3.2.2.4	operator*() [1/2]	9
3.2.2.5	operator*() [2/2]	10
3.2.2.6	operator+()	10
3.2.2.7	operator-()	11
3.2.2.8	operator/()	11
3.2.2.9	operator=()	11
3.2.3	Documentation des fonctions amies et associées	11
3.2.3.1	operator*	12

4	Documentation des fichiers	13
4.1	Référence du fichier CMatrice.hh	13
4.2	Référence du fichier Exception.hh	13
4.2.1	Documentation des macros	13
4.2.1.1	DIVISION_PAR_ZERO	13
4.2.1.2	TAILLE_MATRICE_INVALIDE	13
	Index	15

Chapitre 1

Index des classes

1.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

CException	5
CMatrice< Mtype >	7

Chapitre 2

Index des fichiers

2.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

CMatrice.hh	13
Exception.hh	13

Chapitre 3

Documentation des classes

3.1 Référence de la classe CException

```
#include <Exception.hh>
```

Graphe de collaboration de CException :

Fonctions membres publiques

- [CException](#) ()
Construit un nouvel objet [CException](#) sans paramètre.
- [CException](#) (unsigned int uiType)
Construit un nouvel objet [CException](#) a partir d'un int.
- [CException](#) (string sTxt)
Construit un nouvel objet [CException](#) a partir d'un string.
- unsigned int [uiEXPgetType](#) ()
uiEXPtype's getter
- void [sEXPmessageException](#) ()
retourne un petit texte donnant le type d'exception levé

3.1.1 Documentation des constructeurs et destructeur

3.1.1.1 CException() [1/3]

```
CException::CException ( )
```

Construit un nouvel objet [CException](#) sans paramètre.

3.1.1.2 CException() [2/3]

```
CException::CException (
    unsigned int uiType )
```

Construit un nouvel objet [CException](#) a partir d'un int.

Paramètres

<i>uiType</i>	un int
---------------	--------

3.1.1.3 CException() [3/3]

```
CException::CException (
    string sTxt )
```

Construit un nouvel objet [CException](#) a partir d'un string.

Paramètres

<i>sTxt</i>	Message de l'exception
-------------	------------------------

3.1.2 Documentation des fonctions membres**3.1.2.1 sEXPmessageException()**

```
void CException::sEXPmessageException ( )
```

retourne un petit texte donnant le type d'expection levé

Renvoie

char*

3.1.2.2 uiEXPgetType()

```
unsigned int CException::uiEXPgetType ( )
```

uiEXPtype's getter

Renvoie

unsigned int

La documentation de cette classe a été générée à partir du fichier suivant :

— [Exception.hh](#)

3.2 Référence du modèle de la classe CMatrice< Mtype >

```
#include <CMatrice.hh>
```

Graphe de collaboration de CMatrice< Mtype > :

Fonctions membres publiques

- [CMatrice](#) ()
Construit un nouvel object CMatrice<Mtype> sans parrametre le nombre de ligne et de colonne sont mis à 0, et le tableau à null.
- [CMatrice](#) (Mtype **pMatrice, unsigned int uiNbLigne, unsigned int uiNbColonne)
Construit un nouvel object CMatrice<Mtype> avec tous les parametres possibles.
- [CMatrice](#) ([CMatrice](#)< Mtype > &MATmatrice)
Constructeur de copy.
- void [MATafficherMatrice](#) ()
affiche une matrice dans le terminal
- [CMatrice](#)< Mtype > & [operator=](#) ([CMatrice](#)< Mtype > MATmatrice)
operateur d'affectation si la matrice a gauche était deja alloue, alors elle sera tout d'abord desaloue
- [CMatrice](#)< Mtype > & [operator+](#) ([CMatrice](#)< Mtype > &MATmat)
operateur d'addition Cette operateur ne modifie pas les valeurs des deux operandes
- [CMatrice](#)< Mtype > & [operator-](#) ([CMatrice](#)< Mtype > &MATmat)
operateur de soustraction Cette operateur ne modifie pas les valeurs des deux operandes
- [CMatrice](#)< Mtype > & [operator*](#) (double dVal)
operateur de multiplication entre un CMatrice et un double Cette operateur de modifie pas les valeurs des deux operandes
- [CMatrice](#)< Mtype > & [operator*](#) ([CMatrice](#)< Mtype > &MATmat)
operateur de multiplication entre deux CMatrices Cette operateur ne modifie pas les valeurs des deux operandes
- [CMatrice](#)< Mtype > & [operator/](#) (double dVal)
- [CMatrice](#)< Mtype > & [MATtransposee](#) ()
faire la transposé d'une matrice Le CMatrice d'origine n'est pas modifie

Fonctions membres publiques statiques

- static [CMatrice](#)< double > * [MATlireFichier](#) (char *sChemin)
lit un fichier et retourne un object CMatrice<double>

Amis

- [CMatrice](#)< Mtype > & [operator*](#) (double dVal, [CMatrice](#)< Mtype > &MATmat)
operateur de multiplication entre un double et un CMatrice Cette operateur de modifie pas les valeurs des deux operandes

3.2.1 Documentation des constructeurs et destructeur

3.2.1.1 CMatrice() [1/3]

```
template<class Mtype>
CMatrice< Mtype >::CMatrice ( ) [inline]
```

Construit un nouvel object CMatrice<Mtype> sans parrametre le nombre de ligne et de colonne sont mis à 0, et le tableau à null.

3.2.1.2 CMatrice() [2/3]

```
template<class Mtype>
CMatrice< Mtype >::CMatrice (
    Mtype ** pMatrice,
    unsigned int uiNbLigne,
    unsigned int uiNbColonne ) [inline]
```

Construit un nouvel object CMatrice<Mtype> avec tous les parametres possibles.

Paramètres

<i>pMatrice</i>	un tableau de pointeur de Mtype qui represente les valeurs de la matrice.
<i>uiNbLigne</i>	le nombre de ligne de la matrice
<i>uiNbColonne</i>	le nombre de colonne de la matrice

Exceptions

<i>CException</i>	si le tableau de pointeur est null.
-------------------	-------------------------------------

3.2.1.3 CMatrice() [3/3]

```
template<class Mtype>
CMatrice< Mtype >::CMatrice (
    CMatrice< Mtype > & MATmatrice ) [inline]
```

Constructeur de copy.

Paramètres

<i>MATmatrice</i>	un Object CMatrice<Mtype> à partir duquel creer un nouvel object CMatrice<Mtype>
-------------------	--

3.2.2 Documentation des fonctions membres

3.2.2.1 MATafficherMatrice()

```
template<class Mtype>
void CMatrice< Mtype >::MATafficherMatrice ( ) [inline]
```

affiche une matrice dans le terminal

3.2.2.2 MATlireFichier()

```
template<class Mtype>
static CMatrice<double>* CMatrice< Mtype >::MATlireFichier (
    char * sChemin ) [inline], [static]
```

lit un fichier et retourne un objet CMatrice<double>

Paramètres

<i>sChemin</i>	char*
----------------	-------

Renvoie

CMatrice<double>* un pointeur vers un CMatrice<double> déjà alloué

Exceptions

<i>CException</i>	Si le fichier n'est pas dans le bon format
-------------------	--

3.2.2.3 MATtransposee()

```
template<class Mtype>
CMatrice<Mtype>& CMatrice< Mtype >::MATtransposee ( ) [inline]
```

faire la transposé d'une matrice Le CMatrice d'origine n'est pas modifié

Renvoie

CMatrice<Mtype>& un nouvel objet CMatrice déjà alloué

3.2.2.4 operator*() [1/2]

```
template<class Mtype>
CMatrice<Mtype>& CMatrice< Mtype >::operator* (
    double dVal ) [inline]
```

opérateur de multiplication entre un CMatrice et un double Cette opérateur ne modifie pas les valeurs des deux opérandes

Paramètres

<i>dVal</i>	un double
-------------	-----------

Renvoie

`CMatrice<Mtype>` & un nouvel objet [CMatrice](#) déjà alloué

3.2.2.5 operator*() [2/2]

```
template<class Mtype>
CMatrice<Mtype>& CMatrice< Mtype >::operator* (
    CMatrice< Mtype > & MATmat ) [inline]
```

opérateur de multiplication entre deux `CMatrices` Cette opérateur ne modifie pas les valeurs des deux opérandes

Paramètres

<i>MATmat</i>	un CMatrice
---------------	-----------------------------

Renvoie

`CMatrice<Mtype>` & Un nouvel objet [CMatrice](#) déjà alloué

Exceptions

CException	si les deux matrices n'ont pas les bonnes tailles pour être multipliées
----------------------------	---

3.2.2.6 operator+()

```
template<class Mtype>
CMatrice<Mtype>& CMatrice< Mtype >::operator+ (
    CMatrice< Mtype > & MATmat ) [inline]
```

opérateur d'addition Cette opérateur ne modifie pas les valeurs des deux opérandes

Paramètres

<i>MATmat</i>	une CMatrice
---------------	------------------------------

Renvoie

`CMatrice<Mtype>` & un nouvel objet [CMatrice](#) déjà alloué

Exceptions

CException	si les deux matrices ne sont pas de la même taille
----------------------------	--

3.2.2.7 operator-()

```
template<class Mtype>
CMatrice<Mtype>& CMatrice< Mtype >::operator- (
    CMatrice< Mtype > & MATmat ) [inline]
```

opérateur de soustraction Cette opérateur ne modifie pas les valeurs des deux operands

Paramètres

MATmat	une CMatrice
--------	--------------

Renvoie

CMatrice<Mtype>& un nouvel objet CMatrice déjà alloué

Exceptions

CException	si les deux matrices ne sont pas de la même taille
------------	--

3.2.2.8 operator/()

```
template<class Mtype>
CMatrice<Mtype>& CMatrice< Mtype >::operator/ (
    double dVal ) [inline]
```

3.2.2.9 operator=()

```
template<class Mtype>
CMatrice<Mtype>& CMatrice< Mtype >::operator= (
    CMatrice< Mtype > MATmatrice ) [inline]
```

opérateur d'affectation si la matrice à gauche était déjà allouée, alors elle sera tout d'abord désallouée

Paramètres

MATmatrice	un Object CMatrice
------------	--------------------

Renvoie

CMatrice<Mtype>& un nouvel objet CMatrice déjà alloué.

3.2.3 Documentation des fonctions amies et associées

3.2.3.1 operator*

```
template<class Mtype>
CMatrice<Mtype>& operator* (
    double dVal,
    CMatrice< Mtype > & MATmat ) [friend]
```

opérateur de multiplication entre un double et un [CMatrice](#) Cette opérateur ne modifie pas les valeurs des deux opérandes

Paramètres

<i>dVal</i>	un double
<i>MATmat</i>	un CMatrice correctement alloué et initialisé

Renvoie

[CMatrice<Mtype>&](#) un nouvel objet [CMatrice](#) déjà alloué

Exceptions

CException	si le CMatrice est null
----------------------------	---

La documentation de cette classe a été générée à partir du fichier suivant :

— [CMatrice.hh](#)

Chapitre 4

Documentation des fichiers

4.1 Référence du fichier CMatrice.hh

```
#include <iostream>
#include <fstream>
#include <string>
#include "CMatrice.hh"
#include "Exception.hh"
```

Graphe des dépendances par inclusion de CMatrice.hh :

4.2 Référence du fichier Exception.hh

```
#include <iostream>
#include <string.h>
```

Graphe des dépendances par inclusion de Exception.hh : Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

— class [CException](#)

Macros

— [#define TAILLE_MATRICE_INVALIDE 1](#)
— [#define DIVISION_PAR_ZERO 2](#)

4.2.1 Documentation des macros

4.2.1.1 DIVISION_PAR_ZERO

```
#define DIVISION_PAR_ZERO 2
```

4.2.1.2 TAILLE_MATRICE_INVALIDE

```
#define TAILLE_MATRICE_INVALIDE 1
```


Index

- CException, [5](#)
 - CException, [5](#), [6](#)
 - sEXPmessageException, [6](#)
 - uiEXPgetType, [6](#)
- CMatrice
 - CMatrice, [7](#), [8](#)
 - MAfficherMatrice, [8](#)
 - MATlireFichier, [8](#)
 - MATtransposee, [9](#)
 - operator*, [9–11](#)
 - operator+, [10](#)
 - operator-, [10](#)
 - operator/, [11](#)
 - operator=, [11](#)
- CMatrice< Mtype >, [7](#)
- CMatrice.hh, [13](#)
- DIVISION_PAR_ZERO
 - Exception.hh, [13](#)
- Exception.hh, [13](#)
 - DIVISION_PAR_ZERO, [13](#)
 - TAILLE_MATRICE_INVALIDE, [13](#)
- MAfficherMatrice
 - CMatrice, [8](#)
- MATlireFichier
 - CMatrice, [8](#)
- MATtransposee
 - CMatrice, [9](#)
- operator*
 - CMatrice, [9–11](#)
- operator+
 - CMatrice, [10](#)
- operator-
 - CMatrice, [10](#)
- operator/
 - CMatrice, [11](#)
- operator=
- operator=
 - CMatrice, [11](#)
- sEXPmessageException
 - CException, [6](#)
- TAILLE_MATRICE_INVALIDE
 - Exception.hh, [13](#)
- uiEXPgetType
 - CException, [6](#)