

Rapport du projet :

Projet C++ : Les matrices

Filière : Informatique/DI3

Module : *Mise en œuvre C++/Projet tutoré*

Réalisé par :

EL MAAIDLI Ilyass

DGHOUGH I Oussama

Encadré par :

Mr. Vincent T'kindt

SOMMAIRE

1- Introduction

2- Diagramme de classes

3- Choix des classes

4- Test unitaire

5- Utilisation

6- Annexe

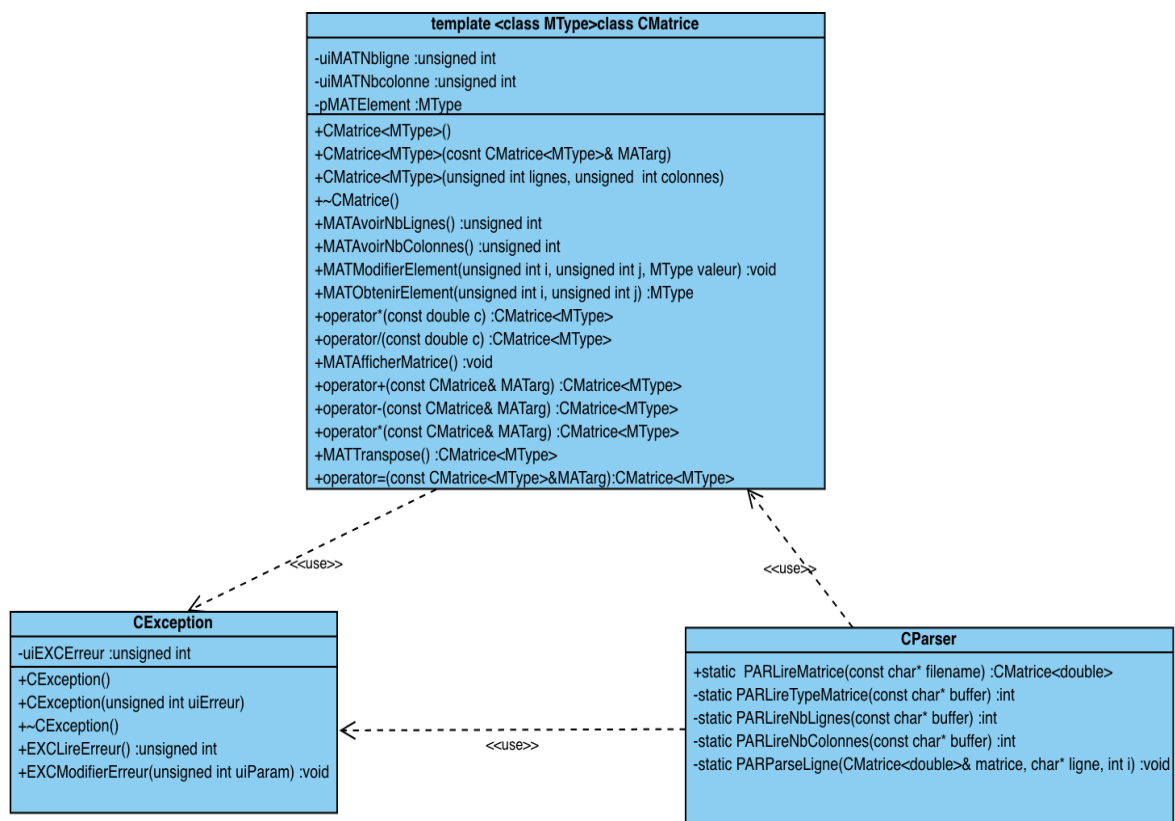
1-Introduction :

Dans ce projet nous devons réaliser une librairie de classes et fonctions permettant de gérer des matrices d'éléments quelconque. Ces matrices doivent être lues depuis des fichiers à l'aide d'une fonction spécifique.

2-Diagramme de classes :

Avant de se lancer pour créer le diagramme de classe, nous avons décidé de créer 3 classes pour ce projet :

- **CMatrice** : Responsable de la création de la matrice ainsi que les opérations simples et complexes (Addition, soustraction, multiplication par un scalaire, multiplication de deux matrices, division par un scalaire). De plus d'une surcharge de l'opérateur =.
- **CParser** : A pour but d'offrir les méthodes nécessaires pour la lecture et la création des matrices à partir des fichiers texte.
- **CException** : Classe classique pour gérer les exceptions.



3-CHOIX DE CLASSES

->**La classe CMatrice** : C'est une classe patron qui a pour objectif la gestion facile et efficace des matrices. Elle utilise un type g n rique *MType* pour d finir le type d' l ment de la matrice. Il contient trois attributs priv s, *uiMATNbLigne* pour indiquer le nombre de lignes, *uiMATNbColonnes* pour indiquer le nombre de colonnes et *pMATElement* comme un tableau bidimensionnel de type *MType* qui contient les  l ments de la matrice.

La classe *CMatrice* contient plusieurs constructeurs et destructeurs utilis s pour cr er (constructeur par d faut) et d truire les instances de la classe(destructeur) ainsi que d'autres constructeurs comme de recopie et de confort. En outre, elle comporte des fonctions permettant de manipuler les  l ments de la matrice, y compris de les  diter et de les visualiser ainsi des accesseurs pour acc der aux donn es de la matrice en dehors de la classe.

De plus, la classe *CMatrice* surcharge les op rateurs de base, tels que l'addition, la soustraction et la multiplication par un scalaire ou par une autre matrice. Elle fournit  galement une m thode pour obtenir la transpos e de la matrice.

En r sum  , la classe *CMatrice* offre une abstraction pratique pour la gestion des matrices , Elle fournit une grande flexibilit , une facilit  d'utilisation et des fonctions avanc es pour traiter matrices .

—->**G nie log** : la classe *CMatrice* respecte le principe de l'encapsulation, ainsi que le principe de l'abstraction, en offrant une interface simple et g n rique pour manipuler les matrices.

De plus, on peut r utiliser cette classe dans d'autres projets, dont la d claration de cette classe en tant que patron permet de cr er des matrices de diff rents types de donn es, sans avoir besoin de r  crire le code pour chaque type.

—>**La classe CParser** : Dans cette classe on a d cid  de d composer le travail principal de cette qui est la lecture et la cr ation des matrices   partir des fichiers en plusieurs m thodes afin de simplifier son utilisation et rendre le fonctionnement plus clair et plus simple. On opte pour l'utilisation de m thodes statiques afin de permettre l'acc s aux fonctions de la classe sans avoir besoin d'instancier un objet de la classe.

En g n ral cette classe contient une m thode publique *PARLireMatrice* pour lire et cr er la matrice   partir du fichier pass  en param tre, qui va appeler tous les autres m thodes *PARLireTypeMatrice*, *PARLireNbLignes*, *PARLireNbColonnes*, *PARParseLigne*. Elles sont d finies comme priv es car on va les utiliser au sein de la classe seulement.

Cela donne une classe simple et facile   utiliser.

—->**G nie log** : Cette classe r pond   plusieurs principes et bonnes pratiques de d veloppement logiciel. Avec le choix qu'on a fait on a rendu le code plus modulaire et plus facile   maintenir. Plus simple et coh rent et valide pour la r utilisation.

--->La class CException : le choix de cette classe se fait pour g rer et identifier les erreurs lorsqu'une exception anormale peut survenir au cours de l'ex cution du programme. Lorsqu'une situation exceptionnelle se produit, la classe *CException* lui attribue un code d'erreur unique auquel il est possible de se r f rer ult rieurement dans le programme, pour cela on a choisi cette classe pour permet aux utilisateurs de la classe *CMatrice* de d tecter et de corriger rapidement les erreurs qui se produisent lors de l'utilisation de la classe.

--->Genie log : la classe *CException* respecte aussi les principes d di s au g nie log afin de bien g rer les erreurs dans le programme pour le rendre plus robuste et plus fiable .Ainsi la r utilisation de cette classe dans plusieurs projets pour g rer les erreurs permet de centraliser cette gestion et facilite le traitement et cela ne qu'une am lioration de la maintenabilit  .Cependant , il est important de bien d finir les exceptions qui peuvent survenir et de la g rer d'une mani re appropri  dans n'importe quel projet utilise cette classe .

4-TESTS UNITAIRES :

Pour tester nos m thodes fonctionnent-elles bien nous avons fait une main provisoire dans le r pertoire de projet appel  "testdesmethodes.cpp".

Il teste les diverses m thodes mises en  uvre dans les classes *CMatrice* et *CParser*. En premier lieu, nous avons lanc  un g n rateur de nombres al atoires   l'aide de la fonction *srand()* et cr   deux matrices al atoires de taille $n*m$ pour tester toutes les m thodes de la classe *CMatrice*. Ainsi on a test  la m thode responsable de la lecture d'une matrice   partir d'un fichier *PARLireMatrice()* par la cr ation d'une instantiation de la classe *CParser* qui est utilis e pour lire une matrice   partir d'un fichier texte nomm  "matrice1.txt" qui se trouve dans le m me emplacement que le fichier "testdesmethodes.cpp". On stocke ensuite la matrice dans une variable de type *CMatrice<double>*. Par la suite, la matrice est affich e   l' cran   l'aide de la m thode *MATAfficherMatrice()*.

Deuxi mement, on ajoute des blocs try-catch pour g rer les exceptions (Annexe) g n r es lors de l'ex cution d'op rations avec des matrices de tailles diff rentes. Finalement, les r sultats de chacune des m thodes ont  t  affich s   l' cran afin de v rifier leur fonctionnement.

N.B: si on veut ex cuter ce fichier de tests il faut mettre en commentaire le fichier de la fonction main principale "matrices.cpp"

```

Nb de lignes de la matrice 1 :
4
Nb de colonnes de la matrice 1 :
4
Affichage matrice 1 :
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0

Affichage matrice 2 :
1 8 2 0
1 3 5 3
9 8 8 9
6 3 9 8

Affichage element(4,4) de la matrice 2 :
8
Affichage element(5,5) de la matrice 2 :
i et j sans hors dimension code num : 11
Somme de matrice 1 et matrice 2 :
1 8 2 0
1 3 5 3
9 8 8 9
6 3 9 8
    Les matrices doivent avoir la meme taille code num : 13
Difference de matrice 1 et matrice 2 :
-1 -8 -2 0
-1 -3 -5 -3
-9 -8 -8 -9
-6 -3 -9 -8
    Les matrices doivent avoir la meme taille code num : 13
transpose de la matrice :
1 1 9 6
8 3 8 3
2 5 8 9
0 3 9 8

```

```

produit de matrice 1 et matrice 2 :
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
    nombre de colonnes de la matrice gauche doit etre egal au nombre de lignes de la matrice droite code num : 14
Division de la mat1 par un scalaire :
0.5 4 1 0
0.5 1.5 2.5 1.5
4.5 4 4 4.5
3 1.5 4.5 4
    l'erreur de Division par zero num : 12
50.5 50.1 50.2
0.7 0.1 3.5
5.23 600.1 50.9

Sortie de C:\Users\lfo9m\Desktop\Spé Info\C++ Project\matrices\Release\matrices.exe (processus 50160). Code : 0.
Pour fermer automatiquement la console quand le débogage s'arrête, activez Outils->Options->Débogage->Fermer automatiquement la console à l'arrêt du débogage.
Appuyez sur une touche pour fermer cette fenêtre. . .

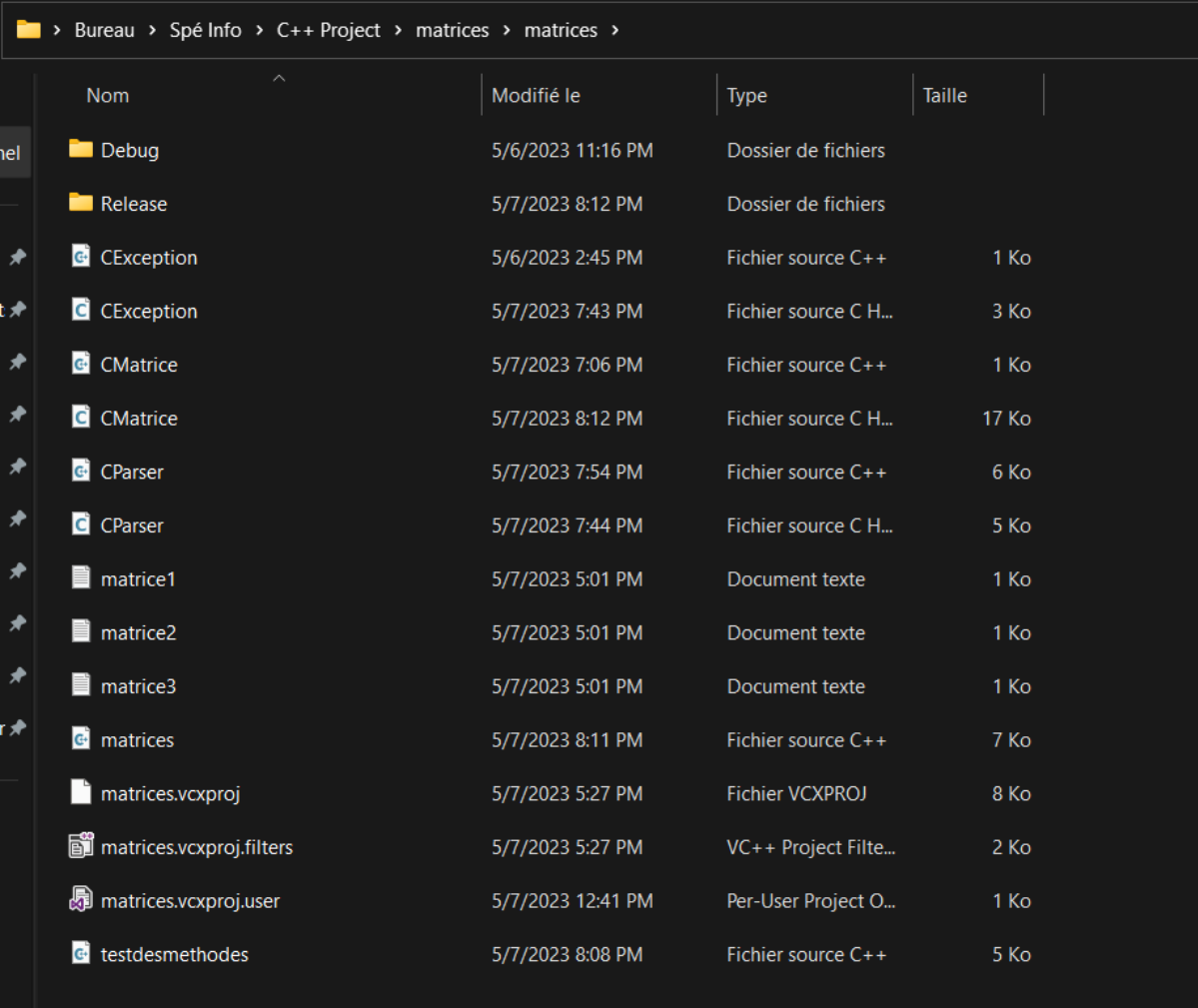
```

5-UTILISATION :

Sur visual studio 2019 :

important : j'ai utilisé 3 fichiers texte qui sont: **matrice1.txt** ; **matrice2.txt** ; **matrice3.txt**

il faut les mettre dans le même dossier que les fichiers .h et .cpp dans le cas d'exécution à partir de Visual studio 2019.

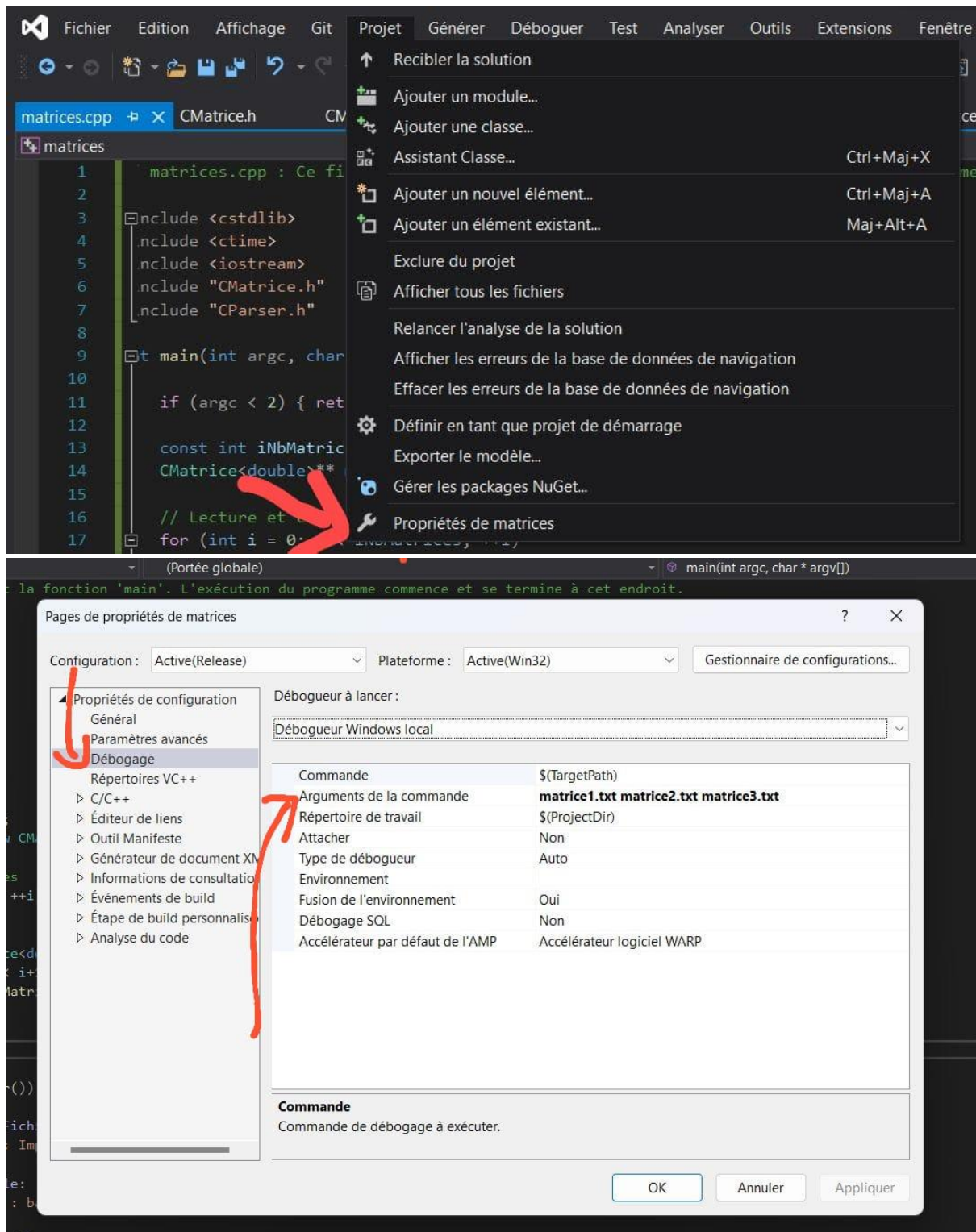


The screenshot shows the File Explorer in Visual Studio 2019, displaying the contents of the 'matrices' project folder. The path is 'Bureau > Spé Info > C++ Project > matrices > matrices'. The table below represents the data shown in the screenshot.

Nom	Modifié le	Type	Taille
Debug	5/6/2023 11:16 PM	Dossier de fichiers	
Release	5/7/2023 8:12 PM	Dossier de fichiers	
CException	5/6/2023 2:45 PM	Fichier source C++	1 Ko
CException	5/7/2023 7:43 PM	Fichier source C H...	3 Ko
CMatrice	5/7/2023 7:06 PM	Fichier source C++	1 Ko
CMatrice	5/7/2023 8:12 PM	Fichier source C H...	17 Ko
CParser	5/7/2023 7:54 PM	Fichier source C++	6 Ko
CParser	5/7/2023 7:44 PM	Fichier source C H...	5 Ko
matrice1	5/7/2023 5:01 PM	Document texte	1 Ko
matrice2	5/7/2023 5:01 PM	Document texte	1 Ko
matrice3	5/7/2023 5:01 PM	Document texte	1 Ko
matrices	5/7/2023 8:11 PM	Fichier source C++	7 Ko
matrices.vcxproj	5/7/2023 5:27 PM	Fichier VCXPROJ	8 Ko
matrices.vcxproj.filters	5/7/2023 5:27 PM	VC++ Project Filte...	2 Ko
matrices.vcxproj.user	5/7/2023 12:41 PM	Per-User Project O...	1 Ko
testdesmethodes	5/7/2023 8:08 PM	Fichier source C++	5 Ko

Comment passer en paramètre les noms des fichiers ?

Aller sur Projet->Propriétés de matrices->Débogage->Ajouter dans la case Argument de la commande les noms des fichiers



Voici un exemple d'exécution :


```

Console de débogage Micros... x + v
matrice1
50.5 50.1 50.2
0.7 0.1 3.5
5.23 600.1 50.9

matrice2
50.5 50.1 50.2
5.23 600.1 50.9
5.23 600.1 50.9

matrice3
0.7 0.1 3.5
5.23 600.1 50.9
5.23 600.1 50.9

Entrez une valeur pour c: 2
Resultat de la multiplication de la matrice 1 par 2 :
101 100.2 100.4
1.4 0.2 7
10.46 1200.2 101.8
Resultat de la multiplication de la matrice 2 par 2 :
101 100.2 100.4
10.46 1200.2 101.8
10.46 1200.2 101.8
Resultat de la multiplication de la matrice 3 par 2 :
1.4 0.2 7
10.46 1200.2 101.8
10.46 1200.2 101.8
Resultat de la division de la matrice 1 par 2 :
25.25 25.05 25.1
0.35 0.05 1.75
2.615 300.05 25.45
Resultat de la division de la matrice 2 par 2 :
25.25 25.05 25.1
2.615 300.05 25.45
2.615 300.05 25.45
Resultat de la division de la matrice 3 par 2 :
0.35 0.05 1.75
2.615 300.05 25.45
2.615 300.05 25.45

```

A Partir du CMD :

important: il faut mettre les fichiers dans le même dossier que l'exécutable « .exe » du projet dans le dossier RELEASE.

Bureau > Spé Info > C++ Project > matrices > Release				
	Nom	Modifié le	Type	Taille
	matrice1	5/7/2023 5:01 PM	Document texte	1 Ko
	matrice2	5/7/2023 5:01 PM	Document texte	1 Ko
	matrice3	5/7/2023 5:01 PM	Document texte	1 Ko
	matrices	5/7/2023 9:55 PM	Application	30 Ko
	matrices.pdb	5/7/2023 9:55 PM	Program Debug D...	892 Ko

Voici un exemple d'exécution :

```

Windows PowerShell
PS C:\Users\lfo9m\Desktop\Spé Info\C++ Project\matrices\Release> dir

Répertoire : C:\Users\lfo9m\Desktop\Spé Info\C++ Project\matrices\Release

Mode                LastWriteTime         Length Name
----                -
-a----          5/7/2023   5:01 PM             108 matrice1.txt
-a----          5/7/2023   5:01 PM             109 matrice2.txt
-a----          5/7/2023   5:01 PM             106 matrice3.txt
-a----          5/7/2023   9:55 PM          30208 matrices.exe
-a----          5/7/2023   9:55 PM          913408 matrices.pdb

PS C:\Users\lfo9m\Desktop\Spé Info\C++ Project\matrices\Release> .\matrices.exe matrice1.txt matrice2.txt matrice3.txt
matrice1
50.5 50.1 50.2
0.7 0.1 3.5
5.23 600.1 50.9

matrice2
50.5 50.1 50.2
5.23 600.1 50.9
5.23 600.1 50.9

matrice3
0.7 0.1 3.5
5.23 600.1 50.9
5.23 600.1 50.9

Entrez une valeur pour c: 2
Resultat de la multiplication de la matrice 1 par 2 :
101 100.2 100.4
1.4 0.2 7
10.46 1200.2 101.8
Resultat de la multiplication de la matrice 2 par 2 :
101 100.2 100.4
10.46 1200.2 101.8
10.46 1200.2 101.8

```

```

Windows PowerShell

Entrez une valeur pour c: 2
Resultat de la multiplication de la matrice 1 par 2 :
101 100.2 100.4
1.4 0.2 7
10.46 1200.2 101.8
Resultat de la multiplication de la matrice 2 par 2 :
101 100.2 100.4
10.46 1200.2 101.8
10.46 1200.2 101.8
Resultat de la multiplication de la matrice 3 par 2 :
1.4 0.2 7
10.46 1200.2 101.8
10.46 1200.2 101.8
Resultat de la division de la matrice 1 par 2 :
25.25 25.05 25.1
0.35 0.05 1.75
2.615 300.05 25.45
Resultat de la division de la matrice 2 par 2 :
25.25 25.05 25.1
2.615 300.05 25.45
2.615 300.05 25.45
Resultat de la division de la matrice 3 par 2 :
0.35 0.05 1.75
2.615 300.05 25.45
2.615 300.05 25.45
Resultat de l'addition de toutes les matrices
101.7 100.3 103.9
11.16 1200.3 105.3
15.69 1800.3 152.7
Resultat de l'operation: M1 - M2 + M3 - M4 + M5 - M6 + ...
0.7 0.1 3.5
0.7 0.1 3.5
5.23 600.1 50.9
Resultat du produit des matrices
370138 4.22236e+07 3.59211e+06
12662.2 1.44853e+06 123053
2.22179e+06 2.54638e+08 2.1611e+07
PS C:\Users\lfo9m\Desktop\Spé Info\C++ Project\matrices\Release> |

```

Fonction main :

Notre fonction main qui a comme nom "matrices.cpp"

Elle prend en paramètre au moins un nom de fichier.

La fonction commence par la création d'un tableau de pointeurs de CMatrice<double> pour stocker les matrices lues et créées à partir des fichiers passés en paramètre.

Ensuite on a réalisé toutes les opérations demandées pour la fonction main.

Pour la gestion des exceptions on a créé des boucles switch pour la détection du type d'exception et l'affichage d'un message pour que l'utilisateur puisse comprendre de quoi s'agit-il le problème.

Voici toutes les exceptions (*voir annexe en bas*) avec leurs messages associés :

-> **EXC_ErreurOuvertureFichier**: "Erreur: Impossible d'ouvrir le fichier "

-> **EXC_BaliseIntrouvable**: "Erreur : balise introuvable dans le fichier "

-> **EXC_TypeMatriceNonDouble**: "Erreur: Type de la matrice (non double) non pris en charge du fichier "

-> **EXC_DimensionInvalide**: "Erreur: Dimension invalide de la matrice dans le fichier "

-> **EXC_AllocationMemoireEchouee**: "Erreur: Allocation de mémoire a échoué pour le produit de la matrice "

-> **EXC_DivisionParZero**: "Erreur: Division par zéro pour la matrice "

-> **EXC_TaillesMatricesNonIdentiques**: "Exception : Tailles des matrices non identiques."

-> **EXC_ErreurHorsLimite**: "Exception : Erreur hors limite"

-> **EXC_ErreurConditionTaillePourMultiplication**: "Exception : Erreur de condition de taille pour la multiplication."

-> Pour toute autre exception levée non connue on affiche "ERROR"

6-ANNEXE:

Numéro du code d'erreur	Classe	Signification
11	CMatrixe	EXC_ErreurHorsLimite
12		EXC_DivisionParZero
13		EXC_TaillesMatricesNonIdentiques
14		EXC_ErreurConditionTaillePourMultiplication
15		EXC_DimensionInvalide
16		EXC_AllocationMemoireEchouee
21	CParser	EXC_ErreurOuvertureFichier
22		EXC_BaliseIntrouvable
23		EXC_TypeMatriceNonDouble