# 2022年计算机图形学作业1

程诗涵 PB19000216 ID：21



# Part 0: 开发环境

Visual Studio 2019

Eigen 3.4.0

# Part 1: 功能简介

在作业1中，我实现了两个矩阵模板类，分别是matrix_base(即一般矩阵)和matrix_sparse(稀疏矩阵)，现在将所有功能列举如下，其中粗体部分表示不同于原框架的新设计或补充功能：

## Part 1.1: 矩阵类的结构

- 对于一般矩阵，封装了矩阵的行数和列数以及存储矩阵元素的指针。
- 对于稀疏矩阵，封装了矩阵的行数和列数以及非零元个数的相关信息，其中对于稀疏矩阵的元素存贮，我利用了STL中的map，具体结构为{key=pair<int,int>,value}

> 实际上在开始写程序的时候我想过先定义一个矩阵的基类，然后定义两个派生类，分别是稠密矩阵和稀疏矩阵的，后来发现其实两类的操作和结构差别都比较大，不太适合，于是选择分开定义。

## Part 1.2: 矩阵的初始化

- 默认初始化：根据图形学的习惯（和Unity中矩阵类似），matrix_base的默认初始化方法是创造一个四阶的单位矩阵，matrix_sparse中则是创造一个四阶的零矩阵。
- 数值初始化方法：提供了初始化任意row*col的零矩阵方法；通过一个列表和row,col值的初始化方法。
- **拷贝构造法**：由于matrix_base中沿用了指针的数据表示方法，因此这里的不能使用默认的拷贝函数，需要自己额外定义提供一个。
- **路径构造法**：对一个给定的路径，可以从该路径中对应的脚本文件文件中读取矩阵，其中脚本文件中对矩阵的定义规则是第一行为矩阵的行数和列数，后几行是矩阵的各个元素，每一行相邻元素之间用空格隔开。
- **不同矩阵类型之间的相互转化构造**：通过将两类分别设置成另外一类的友元类（friend class）的方法，可以将普通矩阵和稀疏矩阵互相转化构造。

## Part 1.3：矩阵的运算操作（运算符号的重载）

- 一般矩阵：在matrix_base中定义了加减乘除和矩阵的标量操作，处理的较为不错的一点是利用Guass消元法进行**矩阵求逆**操作，从而达到对可逆矩阵的求逆操作。
- 对于稀疏矩阵来说，为了节省操作时间，对于矩阵的加法和减法就是对应的两个矩阵的合并，时间复杂度为$\mathcal{O}(m)$，其中m为稀疏矩阵的非零元数目，一般来说稀疏矩阵有$m = o(n)$；对于稀疏矩阵的乘法和除法，为了方便我的实现算法就是先将稀疏矩阵转化成普通矩阵，对普通矩阵进行相应的操作后，再转化回稀疏矩阵。

## Part 1.4: 鲁棒性处理

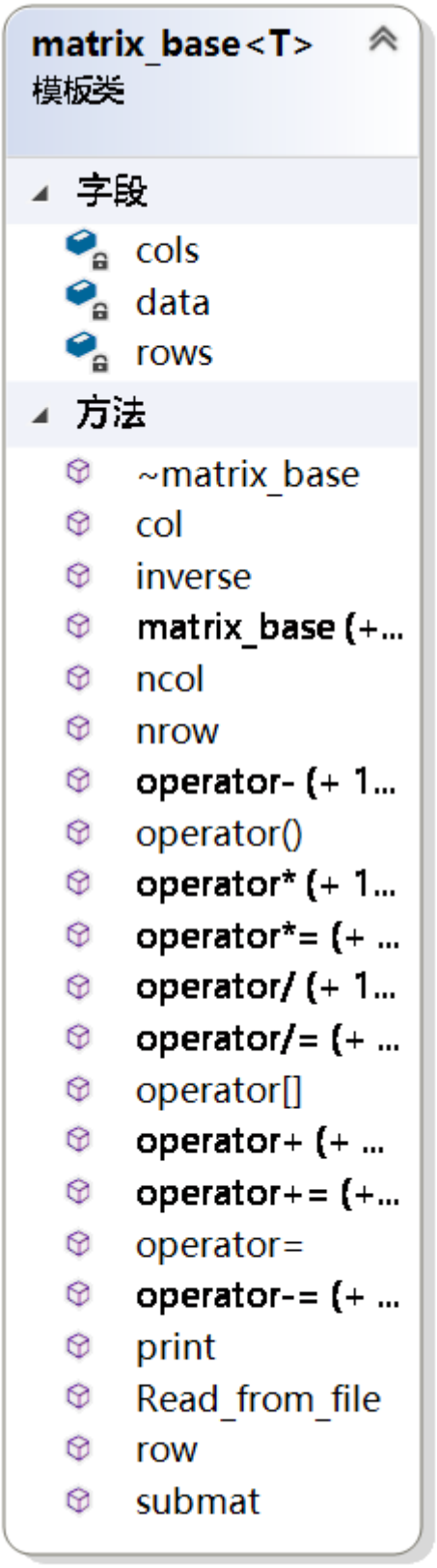
对于矩阵的运算操作，有时因为用户的输入或者操作的原因，会出现类似于两个不同维数的矩阵的相加操作，或者矩阵无法相乘的情况；另一方面，由于程序设计中涉及了矩阵的除法（求逆操作），因此我们的程序需要先对求逆矩阵的可逆性进行检查，避免NA现象出现，在我的实现中，我采取了以下的代码设计：

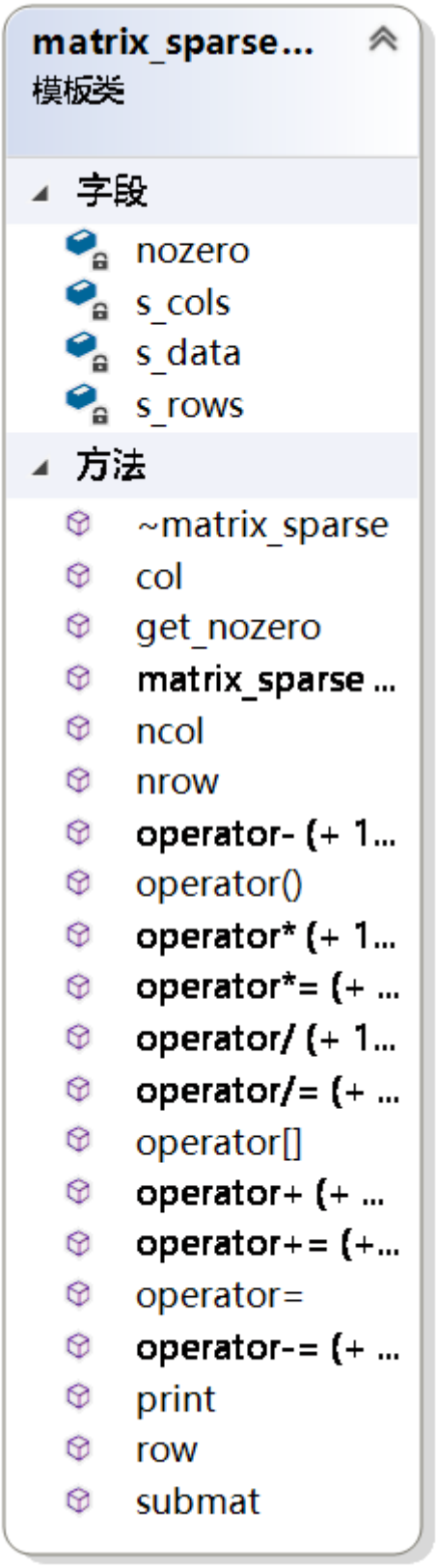
- 对于矩阵的维数合理性判断，统一使用assert函数。
- 对于矩阵的求逆操作，使用try-catch的组合，对于奇异矩阵，提醒用户该矩阵不可进行除法运算。

# Part 2: 实验结果

| | matrix_base | Eigen |
|---|---|---|
| 矩阵A | $\begin{pmatrix} 4 & 1846 & 633 & 2650 \\ 1916 & 1572 & 1147 & 2935 \\ 2696 & 2446 & 570 & 2814 \\ 2328 & 1682 & 996 & 49 \end{pmatrix}$ | $\begin{pmatrix} 4 & 1846 & 633 & 2650 \\ 1916 & 1572 & 1147 & 2935 \\ 2696 & 2446 & 570 & 2814 \\ 2328 & 1682 & 996 & 49 \end{pmatrix}$ |
| 矩阵B | $\begin{pmatrix} 299 & 1194 & 482 \\ 543 & 3239 & 1460 \\ 390 & 15 & 29 \\ 1238 & 1742 & 1871 \end{pmatrix}$ | $\begin{pmatrix} 299 & 1194 & 482 \\ 543 & 3239 & 1460 \\ 390 & 15 & 29 \\ 1238 & 1742 & 1871 \end{pmatrix}$ |
| 矩阵C=A*B | $\begin{pmatrix} 4.53114e+06 & 1.06098e+07 & 7.6736e+06 \\ 5.50734e+06 & 1.25094e+07 & 8.74328e+06 \\ 5.84031e+06 & 1.60522e+07 & 1.01522e+07 \\ 2.0585e+06 & 8.32793e+06 & 3.69838e+06 \end{pmatrix}$ | $\begin{pmatrix} 4.53114e+06 & 1.06098e+07 & 7.6736e+06 \\ 5.50734e+06 & 1.25094e+07 & 8.74328e+06 \\ 5.84031e+06 & 1.60522e+07 & 1.01522e+07 \\ 2.0585e+06 & 8.32793e+06 & 3.69838e+06 \end{pmatrix}$ |
| 运算+打印时间 | 0.021677 s | 0.0287623 s |
| I/A | $\begin{pmatrix} -0.000487767 & 0.000158693 & 0.000294542 & -4.13191e-05 \\ 0.000558773 & -0.000708742 & 0.000207044 & 0.000342579 \\ 0.000199342 & 0.000811226 & -0.00104324 & 0.000540147 \\ -5.87649e-05 & 0.000299697 & 0.000104524 & -0.000367603 \end{pmatrix}$ | $\begin{pmatrix} -0.000487767 & 0.000158693 & 0.000294542 & -4.13191e-05 \\ 0.000558773 & -0.000708742 & 0.000207044 & 0.000342579 \\ 0.000199342 & 0.000811226 & -0.00104324 & 0.000540147 \\ -5.87649e-05 & 0.000299697 & 0.000104524 & -0.000367603 \end{pmatrix}$ |

# Part 3: 程序类图




## Part 4: 总结

通过本次作业，夯实了同学对面向对象编程的理解，了解了C++的特性。