

CS205 C/ C++ Program Design

Assignment 4

姓名:庾立轩








学号:3122001509

本次作业 4 中有三个文件:

1. main.cpp-->主程序
2. matrix.hpp-->矩阵的头文件
3. matrix.cpp-->矩阵的实现文件

注 1:本次程序旨在表现函数效果, 因此缺乏部分错误检查, 而程序实现可能比较粗糙, 望老师见谅。

已上传 github

| | | | |
|---|------------------------------------|--------------|---|
|  | rockyulx Add files via upload ... | now |  2 |
|  | CMakeLists... Add files via upload | now | |
|  | README.md Initial commit | 1 minute ago | |
|  | main.cpp Add files via upload | now | |
|  | matrix.cpp Add files via upload | now | |
|  | matrix.hpp Add files via upload | now | |

代码部分:

Matrix 类的基本信息:

构造函数和析构函数我直接写在了头文件里。

```

class Matrix
{
private:
    int rows;
    int cols;
    float *data;
    static char count;

public:
    Matrix()
    {
        rows = 0;
        cols = 0;
        data = nullptr;
    }
    Matrix(int rows, int cols)
    {
        this->rows = rows;
        this->cols = cols;
        this->data = new float[rows * cols]{0};
        count++;
        std::cout << "Matrix " << count << " is created!" << std::endl;
    }
    ~Matrix()
    {
        std::cout << "Matrix " << count << " is deleted!" << std::endl;
        count--;
        delete[] data;
    }
}

```

包含行、列以及指向矩阵位置的指针
还有一个静态变量用于计数

这里构造函数用于初始化数据
而下面的构造函数用于表示行列已经
录入并开辟了一个空间用于存放矩阵

析构函数用于释放空间

用于获取受保护的成员信息的函数

```

int getrows() const;
int getcols() const;
float * getdata() const;

```

运算符输出<<和录入>>还有=,以及矩阵乘法运算的实现函数

```

Matrix &operator=(const Matrix &mat);
friend std::ostream &operator<<(std::ostream &os, const Matrix &mat);
friend std::istream &operator>>(std::istream &is, Matrix &mat);
Matrix operator*(const Matrix &mat) const;
friend Matrix operator*(const Matrix &mat, int x);
friend Matrix operator*(int x, const Matrix &mat);

```

主函数部分的代码:

```

int main()
{
    int x;
    Matrix A(2, 3);
    std::cin >> A;
    std::cout << "Please enter a number:" << std::endl;
    std::cin >> x;
    A = A * x;
    Matrix B(3, 4);
    std::cin >> B;
    Matrix C = A * B;

    std::cout << "A = " << std::endl
              << A << std::endl;
    std::cout << "B = " << std::endl
              << B << std::endl;
    std::cout << "C = " << std::endl
              << C << std::endl;

    system("pause");
    return 0;
}

```

编译和运行:

使用了 cmake 进行编译:

```

[main] Building folder: ALL
[build] Starting build
[proc] Executing command: "C:\Program Files\CMake\bin\cmake.EXE" --build d:/ALL/build --config Release --target all -j 22 -
[build] [100%] Built target matrix
[build] Build finished with exit code 0

```

程序运行:

```
PS D:\ALL\build> .\matrix.exe
```

```
Matrix A is created!
```

```
1.2 1.2 1.2 2.4 2.4 2.4
```

```
Please enter a number:
```

```
3
```

```
Matrix B is created!
```

```
Matrix B is deleted!
```

```
Matrix B is created!
```

```
8 8 8 8 8 6 6 6 6 6 6
```

```
Matrix C is created!
```

```
Matrix C is deleted!
```

```
A =
```

```
3.6 3.6 3.6
```

```
7.2 7.2 7.2
```

```
B =
```

```
8 8 8 8
```

```
8 8 6 6
```

```
6 6 6 6
```

```
C =
```

```
79.2 79.2 72 72
```

```
158.4 158.4 144 144
```

```
请按任意键继续. . .
```

```
Matrix B is deleted!
```

```
Matrix A is deleted!
```

```
Matrix @ is deleted!
```

```
PS D:\ALL\build>
```

程序运行后显示"Matrix A"被创造出来了，我手动录入了6个数字

这里录入一个标量，与矩阵A相乘。

因为矩阵与标量相乘的函数里生成了一个中转的矩阵，因此这里显示有一个矩阵被创造出来而后迅速删除

手动录入第二个矩阵内的数据

此处也是矩阵相乘生成的临时矩阵

进行打印，因为我们录入的标量是3，所以A大了3倍

在程序结束运行后，剩余的内存被释放。

这里出现了一个越界的问题。我判断是在生成C矩阵的时候我用的是" $C = A * B$ "，这样使用之后计数器不会加1，但是在系统结束运行后的内存释放却会让计数器减1，导致显示了字符'A'前一位的字符，目前没有想到该如何解决

总结:

在本次作业 4 中, 我对 C/C++ 运算符的重载的使用更加熟练, 同时也学会了很

多知识点:

1. C++ 语言中类运算符重载的实现;
2. C++ 语言中友元运算符的重载的实现;
3. 内存的管理;
4. 对类的一些理解;

这里向云动力工作室以及老师道个歉, 寒假的放纵导致了现在学习进度比同期人慢太多了, 不过我现在也在积极追赶.