lttzz / **Course-Design**

Branch: **master** ▾ **Course-Design** / T4 / **4.cpp** Find file Copy path

lttzz 矩阵类 d921ed4 2 days ago

1 contributor

135 lines (117 sloc) 2.11 KB

```
/*****************************************************************************
        > File Name:            4.cpp
        > File Category: Course Design
        > Author:               lttzz
        > Mail:                         3344517687@qq.com
 ****************************************************************************/
/*
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
5 5 5 5
6 6 6 6

5       5       5       5
7       7       7       7
9       9       9       9

-3      -3      -3      -3
-3      -3      -3      -3
-3      -3      -3      -3
*/

#include <iostream>
#include <fstream>
#include <cstdio>
using namespace std;

class Matrix {
private:
        int **p;
        int n, m;

public:
        Matrix(int a, int b): n(a), m(b)
        {
                p = new int *[n];
                for (int i = 0; i < n; i++)
                {
                        p[i] = new int [m];
                }
        }
        ~Matrix()
        {
                for (int i = 0; i < n; i++)
                {
                        delete []p[i];
                }
                delete []p;
        }
        void operator = (Matrix& t)
        {
                for (int i = 0; i < t.n; i++)
                {
                        for (int j = 0; j < t.m; j++)
                        {
                                p[i][j] = t.p[i][j];
                        }
                }
        }
        Matrix& operator + (Matrix& t)          //加减操作注意不要直接在数据上修改
        {
                static Matrix M(t.n, t.m);              //定义静态变量作为临时变量,并用'+'的"第二操作数"对其进行初始化
                for (int i = 0; i < t.n; i++)
                {
                        for (int j = 0; j < t.m; j++)
                        {
                                M.p[i][j] = p[i][j] + t.p[i][j];
                        }
                }

                return M;
        }
        Matrix& operator - (Matrix& t)
        {
                static Matrix M(t.n, t.m);
                for (int i = 0; i < t.n; i++)
                {
                        for (int j = 0; j < t.m; j++)
                        {
                                M.p[i][j] = p[i][j] - t.p[i][j];
                        }
                }

                return M;
        }

        friend istream& operator >> (istream& is, Matrix& t);   //提取和插入运算符重载为友元函数
        friend ostream& operator << (ostream& os, Matrix& t);
};


int main(void)
{
        //freopen("./in.txt", "r", stdin);
        Matrix m1(3, 4), m2(3, 4), m3(3, 4), m4(3, 4);

        cin >> m1 >> m2;
        m3 = m1 + m2;
        m4 = m1 - m2;

        cout << m3 << endl << m4;
        ofstream of("matrixs.txt");
        of << m3  << endl << m4;
        return 0;
}


istream& operator >> (istream& is, Matrix& t)
{

        for (int i = 0; i < t.n; i++)
        {
                for (int j = 0; j < t.m; j++)
                {
                        is >> t.p[i][j];
                }
        }

        return is;
}

ostream& operator << (ostream& os, Matrix& t)
{
        for (int i = 0; i < t.n; i++)
        {
                for (int j = 0; j < t.m; j++)
                {
                        os << t.p[i][j] << '\t';
                }
                os << endl;
        }
        return os;
}
```