

d921ed4 2 days ago

1 contributor

135 lines (117 sloc) 2.11 KB

```
1  /*****
2      > File Name:      4.cpp
3      > File Category: Course Design
4      > Author:      lttzz
5      > Mail:      3344517687@qq.com
6      *****/
7  /*
8  1 1 1 1
9  2 2 2 2
10 3 3 3 3
11 4 4 4 4
12 5 5 5 5
13 6 6 6 6
14
15 5      5      5      5
16 7      7      7      7
17 9      9      9      9
18
19 -3      -3      -3      -3
20 -3      -3      -3      -3
21 -3      -3      -3      -3
22 */
23
24 #include <iostream>
25 #include <fstream>
26 #include <cstdio>
27 using namespace std;
28
29 class Matrix {
30 private:
31     int **p;
32     int n, m;
33
34 public:
35     Matrix(int a, int b): n(a), m(b)
36     {
37         p = new int *[n];
38         for (int i = 0; i < n; i++)
39         {
40             p[i] = new int [m];
41         }
42     }
43     ~Matrix()
44     {
45         for (int i = 0; i < n; i++)
46         {
47             delete []p[i];
48         }
49         delete []p;
50     }
51     void operator = (Matrix& t)
52     {
53         for (int i = 0; i < t.n; i++)
54         {
55             for (int j = 0; j < t.m; j++)
56             {
57                 p[i][j] = t.p[i][j];
58             }
59         }
60     }
61     Matrix& operator + (Matrix& t)      //加减操作注意不要直接在数据上修改
62     {
```

```
63         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
64         for (int i = 0; i < t.n; i++)
65         {
66             for (int j = 0; j < t.m; j++)
67             {
68                 M.p[i][j] = p[i][j] + t.p[i][j];
69             }
70         }
71
72         return M;
73     }
74     Matrix& operator - (Matrix& t)
75     {
76         static Matrix M(t.n, t.m);
77         for (int i = 0; i < t.n; i++)
78         {
79             for (int j = 0; j < t.m; j++)
80             {
81                 M.p[i][j] = p[i][j] - t.p[i][j];
82             }
83         }
84
85         return M;
86     }
87
88     friend istream& operator >> (istream& is, Matrix& t); //提取和插入运算符重载为友元函数
89     friend ostream& operator << (ostream& os, Matrix& t);
90 };
91
92
93 int main(void)
94 {
95     //freopen("./in.txt", "r", stdin);
96     Matrix m1(3, 4), m2(3, 4), m3(3, 4), m4(3, 4);
97
98     cin >> m1 >> m2;
99     m3 = m1 + m2;
100    m4 = m1 - m2;
101
102    cout << m3 << endl << m4;
103    ofstream of("matrixs.txt");
104    of << m3 << endl << m4;
105    return 0;
106 }
107
108
109 istream& operator >> (istream& is, Matrix& t)
110 {
111
112     for (int i = 0; i < t.n; i++)
113     {
114         for (int j = 0; j < t.m; j++)
115         {
116             is >> t.p[i][j];
117         }
118     }
119
120     return is;
121 }
122
123 ostream& operator << (ostream& os, Matrix& t)
124 {
125     for (int i = 0; i < t.n; i++)
126     {
127         for (int j = 0; j < t.m; j++)
128         {
129             os << t.p[i][j] << '\t';
130         }
131         os << endl;
132     }
133     return os;
134 }
```