第四次作业

第一题

可以声明在private，无this指针。

第二题

```
#include <iostream>
using namespace std;
class People {
public:

    class Date {
    public:
        Date(int y,int m,int d):year(y),month(m),day(d) {
        }

        ~Date() {
        }
        int year, month, day;
        friend ostream& operator <<(ostream& o, Date& d);
    private:

    };

    char name[11];
    char number[7];
    char sex[3];
    char id[16];
    Date birthday;
```

```
    People(const char name[11], const char number[7], const char sex[3],const char id[16],int y,int m,int d):birthday(y,m,d) {
        strcpy_s(this->name, 11,name);
        strcpy_s(this->number, 7,number);
        strcpy_s(this->sex, 3,sex);
        strcpy_s(this->id, 16,id);
    }

    ~People() {
    }
    People(const People& r) :birthday(r.birthday) {
        strcpy_s(this->name, 11,r.name);
        strcpy_s(this->number,7, r.number);
        strcpy_s(this->sex,3, r.sex);
        strcpy_s(this->id,16, r.id);
    }
    inline void show() {
        cout << "name: " << name << "  " << "number: " << number << "  " << "sex: " << sex << "  " << "id:  " << id << "  " << "birthday:  "<<birthday;
    }
private:

};

int main()
{
    People p("alice", "1", "wo", "12345678",2003, 1, 1);
    People p1(p);
    p1.show();
}

ostream& operator<<(ostream& o, People::Date& d) {
    o << d.year << "年 " << d.month << "月 " << d.day << "日 ";
    return o;
}

```

```
name: alice  number: 1  sex: wo  id:  12345678  birthday:  2003年 1月 1日
```

第三题

```
#include <iostream>
#include <iomanip>
using namespace std;
class Matrix {
public:
    Matrix(int rol,int col,int**& array) {
        init(rol,col);
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                a[i][j] = array[i][j];
            }
        }

    }
    Matrix(int rol,int col){
        init(rol, col);
    }
    Matrix(int rol, int col,int val) {
        init(rol, col);
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                a[i][j] = val;
            }
        }
    }
    ~Matrix() {
    }
    inline void init(int rol,int col){
        m = rol, n = col;
        a = new int* [m];
        for (int i = 0; i < m; i++) {
            a[i] = new int[n];
        }
    }
    inline void set(int x, int y,int val) {
        a[x][y] = val;
    }
    inline int get(int x, int y) {
        return a[x][y];
    }
```

```
41      inline void show() {
42          for (int i = 0; i < m; i++) {
43              for (int j = 0; j < n; j++) {
44                  cout << setw(3) <<setiosflags(ios::left) << a[i]
[j];
45              }
46              cout << endl;
47          }
48          cout << endl << endl;
49      }
50      friend Matrix operator+(Matrix& a, Matrix& b);
51      friend Matrix operator-(Matrix& a, Matrix& b);
52  private:
53      int** a;
54      int m,n;
55  };
56
57
58  int main() {
59      int size;
60      cout << "input the size:";
61      cin >> size;
62
63      int** a =new int*[size];
64      for (int i = 0; i < size; i++) {
65          a[i] = new int[size];
66          for (int j = 0; j < size; j++) {
67              a[i][j] = i + j;
68          }
69      }
70      Matrix ma(size,size,a);
71
72      int** b = new int* [size];
73      for (int i = 0; i < size; i++) {
74          b[i] = new int[size];
75          for (int j = 0; j < size; j++) {
76              cin >> b[i][j];
77          }
78      }
79      Matrix mb(size, size, b);
80
81      ma.show();
82      mb.show();
```

```
    cout << "+" << endl;
    (ma + mb).show();
    cout << "-" << endl;
    (ma - mb).show();
}
Matrix operator-(Matrix& a, Matrix& b) {
    int rol = a.m, col = a.n;
    Matrix m(rol, col);
    for (int i = 0; i < rol; i++) {
        for (int j = 0; j < col; j++) {
            m.set(i, j, (a.get(i, j) - b.get(i, j)));
        }
    }
    return m;
}
Matrix operator+(Matrix& a, Matrix& b) {
    int rol = a.m, col = a.n;
    Matrix m(rol, col);
    for (int i = 0; i < rol; i++) {
        for (int j = 0; j < col; j++) {
            m.set(i, j, (a.get(i, j) + b.get(i, j)));
        }
    }
    return m;
}

```

```
input the size:3
1 2 3
4 5 6
7 8 9
0  1  2
1  2  3
2  3  4


1  2  3
4  5  6
7  8  9


+
1  3  5
5  7  9
9  11 13


-
-1 -1 -1
-3 -3 -3
-5 -5 -5
```