

Homework #16_1

以 c++ 程式編寫，使其具以下功能

1. 利用運算子多載方式，執行矩陣相加，加法符號 + 定義如下

2. 例如：
$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}, A + B = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} \\ a_{21} + b_{21} & a_{22} + b_{22} \end{bmatrix}$$

3. 運算子多載方式請參考 [子目錄 t15_7\(Point2D\)](#)

參考案：共三個檔案及一個專案檔

Matrix.h, Matrix.cpp, hw16_1.cpp & Matrix.dev

Matrix.h

```
Matrix.cpp | Matrix.h | hw16_1.cpp |
1 class Matrix {
2 public:
3     Matrix();
4     Matrix(double, double, double, double);
5     double a() {return _a;}
6     double b() {return _b;}
7     double c() {return _c;}
8     double d() {return _d;}
9     Matrix operator+(const Matrix&); // 重載+運算子
10
11 private:
12     double _a, _b, _c, _d;
13 };
14
```

Matrix.cpp

```
Matrix.cpp | Matrix.h | hw16_1.cpp
1  #include "Matrix.h"
2
3  Matrix::Matrix() {
4      _a = 0;
5      _b = 0;
6      _c = 0;
7      _d = 0;
8  }
9
10 Matrix::Matrix(double a, double b, double c, double d) {
11     _a = a;
12     _b = b;
13     _c = c;
14     _d = d;
15 }
16
17 Matrix Matrix::operator+(const Matrix &p) {
18     Matrix tmp(_a + p._a, _b + p._b, _c+p._c, _d+p._d);
19     return tmp;
20 }
21
```

hw16_1.cpp

```
Matrix.cpp Matrix.h hw16_1.cpp
1  #include <iostream>
2  #include "Matrix.h"
3  using namespace std;
4
5  int main() {
6      Matrix A(1,1,1,1), B(2,2,2,2), C;
7
8      C = A+B;
9      cout << "C =" << endl;
10     cout << C.a() << "," << C.b() << endl;
11     cout << C.c() << "," << C.d() << endl;
12
13     return 0;
14 }
15
```

執行結果

```
J:\Documents and Settings\mahsiang\桌面\課C12\
C =
3.3
3.3
-----
Process exited with return value 0
Press any key to continue . . .
```