### C++二维动态数组的创建与删除

C++中用new动态创建二维数组的格式一般是这样：

TYPE (\*p)[N] = new TYPE [][N];

其中，TYPE是某种类型，N是二维数组的列数。采用这种格式，列数必须指出，而行数无需指定。在这里，p的类型是TYPE\*[N]，即是指向一个有N列元素数组的指针。

还有一种方法，可以不指定数组的列数：

int \*\*p;  
p = new int\*[10];    //注意，int\*[10]表示一个有10个元素的指针数组  
for (int i = 0; i != 10; ++i)  
{  
    p[i] = new int[5];  
}

这里是将p作为一个指向指针的指针，它指向一个包含10个元素的指针数组，并且每个元素指向一个有5个元素的数组，这样就构建了一个10行5列的数组。

当数组使用完毕，释放空间的代码是：

for(int i = 0; i != 5; i++)  
{  
    delete[] p[i];  
}  
 delete[] p;

处理二维数组，可以用降维或是二维法。  
降维法是用一位数组来接受二维数组，将二维元素的首地址&a[0][0]作为参数，传递给函数，函数用int \*接受。  
二维法就直接用二维数组来接受，但是需要指定列数。

如要想创建一个[m][n]的二维数组。

下面为通过动态创建一个指针数组的方法来动态创建二维数组的方法。

**double \*\*data;**

**data = new double\*[m];** //设置行 或直接double \*\*data=new double\*[m]; 一个指针指向一个指针数组。

**for(int j=0;j<m;j++)  
{**  
**data[j] = new double[n];**        //这个指针数组的每个指针元素又指向一个数组。  
**}**

for (int i=0;i<m;i++)

{  
   for (int j=0;j<n;j++)  
   {  
    data[i][j]=i\*n+j;//初始化数组元素  
   }

}

**for (i=0;i<m;i++)  
{  
 delete[] data[i]; //先撤销指针元素所指向的数组  
}                       
delete[] data**;

这种方法是通过先动态创建一个指针数组，然后为指针数组的每个元素再动态指向一个数组的办法来完成的。其创建过程与销毁过程两样重要。

在销毁的过程，先销毁指针数组每个元素指向的数组，然后再销毁这个指针数组。