第九&十部分　内联函数 & 命名空间

1. 预处理　宏定义 #define　被替换名　替换值

为了提升一些简单操作在程序中的效率和减少资源开销，C中可以将一些简单操作定义为宏

#define f(x) (x)*(x) [求平方操作]

本质上是在程序中进行字符串替换，

与函数有本质上的不同 ↓ 在编译过程中

发展 ↓

2. CPP中的内联函数

内联函数的用途类似于宏，但其是真正的函数，需要在声明和定义时使用 inline 关键字

① inline 函数

不能包含递归(一般)

② inline 函数

务必进行声明&定义

为编译器替换做准备

指在编译过程中，

编译器会在内联函数的使用处，

用内联函数代码段替换函数名

而不进入函数栈，即宏展开

特别的，C++中所有在类内定义的函数都为内联函数

3. 组合类

指在类的定义中，成员变量使用其它类作为基本类型，反映了类和类之间的关系

{ 外层类的对象称为对象

1 内层类的对象称为子对象

注：子对象的构造只能在对象构造函数的初始化列表中进行→或为子对象给定无参构造函数

```
class mm{ //内类
int i;
mm(int a):i(a){};
};
```

```
class Wmm{ //类
mm q, s;
Wmm(int a, int b):q(a), s(b){};
};
```

对象在构建时，会先访问初始化列表，之后对子对象进行构建，再访问函数体

4. 命名空间 namespace 封装性

对于一些全局变量、函数等全局内容，其名字是其唯一标识，这导致了大项目可能出现重名现象。

↓

因此，定义一块命名空间 /namespace/，用户可以在该命名空间任意书写代码并由命名空间封装，在全局函数中进行调用。

```
namespace DB{
//code
}
DB::xxx;//由此调用
```

命名空间与类有何不同？

1. 命名空间必须定义在全局域中（对全局域分块封装）
2. 可使用 namespace Lib=MyLib 重命名
3. 命名空间定义尾无需";"符号

※一般来说，一块命名空间中放置的是一类的代码，进行使用
※可以使用语句 using namespace xxx; 来缺省调用 xxx 空间