第八部分 常量 Constants 关键字 const

1. 常量在变量、表达式中的应用

可用 const 关键字对 指针、变量、函数返回值等的数据类型进行定义，生成不可修改的变量

只读

例：

变量 const int X=10 ⟷ int const X=10

注：常变量在声明时务必初始化，因其之后无法赋值。

语法中默认修饰其左侧的数据类型

指针 const int *p=50 ⟷ int const *p=50

注：此处 const 修饰 int，即用户不能通过该指针修改当前地址的值。

但该指针变量 p 不是常量，可以改变其指向区域

int * const p=50

注：此处 const 修饰 int*，即用户不能修改指针变量指向的区域，但可以改变该区域的值

数组 const int a[]={} ⟷ int const a[]={}

注：此处表示数组中每个元素都不可修改

静态，不依赖对象

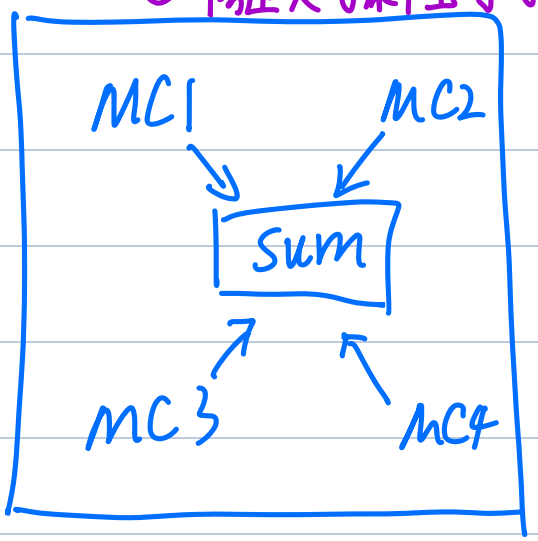
2. Static 类型在类中的使用 [类似全局变量]

static 变量

在类的 private 部分中，定义 static 类型的数据表示该数据为所有该类的对象所共享、修改

其在初始化时，需在类外初始化 类名指定，防止类对象产生时对其重写

```
class MC{
private:
    static int sum;
    ⋮
};
类外初始化 int MC::sum=10;
```



在类中使用 static 类型定义函数时，将获得静态函数。

静态函数无法通过对象访问成员变量，因此需要将

不传递 this 指针（指向当前对象的指针）

不能对象.函数调用 只能类::函数调用

传该访问的对象写入参数列表来间接访问

也正因如此，static 类型的函数与全局函数性质类似 把对象当参数传入

3. const 类型在类中的应用

① 类中的 const 变量必须在初始化列表中被定义

在函数体中会被视为赋值

例：

```
class A{
public:
```

```
A(int i); 声明构造函数
private:
  const int a;
  const int& r;
};
A::A(int i): a(i), r(a)
{
  return ;
}
```

const int a = 11;
对于CPP11，若将新常量初始化给const变量，可在此处进行

初始化列表

构造函数定义

② const 类型函数 → 常函数

这些函数将不允许对类内任一成员变量进行修改，只能读。 只读函数

```
int Year( ) const { 函体 };
```

const修饰符写在最后，表示const修饰函数

但可以将一个变量定义为 mutable 类型，标志该变量可在const类型函数中被修改

```
mutable d; ⇒ int Year( ) const { d++; return d; }
```