

深入理解 static, const, extern



华中科技大学

- static 和 const 作用域
- 变量定义
- 变量声明

华中科技大学 计算机学院

金良海 2022.09





1. static, const 作用域

static、const 对象的作用域是当前模块（当前.cpp文件）。
不同的模块可以定义同名的static、const 变量。

1.cpp

```
const int x = 1;  
static int y = 2;  
static const int z = 3;  
const int a = 5;  
extern int b = 1;  
int k = 0;
```

2.cpp

```
const int x = 1;           //对  
static int y = 2;          //对  
static const int z = 3;    //对  
int k = 0;                 //错误  
extern const int a;        //对  
extern int b = 0;          //错,重复定义  
int main( ) {  
    return x + y + z + a;  
    //错,找不到a  
}
```





2. 变量定义 (1)

变量定义

变量定义 (需要分配内存), 变量定义的2种方法:

- 没有使用extern限定, 可以不初始化。

例如, `const int k = 0;`

`int x, y;`

`A a;` //A定义了无参构造函数

- 使用extern限定, 这时**必须初始化** (不初始化表明仅仅是声明, 这个变量实际上可能不存在)。

`extern const int k = 0;`

`extern int x = 1;`

`extern A a;` //不是定义, 仅是声明

`extern A a(...);` //A定义了有参构造函数

`extern A a = A();` //A定义了无参构造函数





2. 变量定义 (2)

变量定义

变量定义有3种形式:

- 只定义不声明
- 先声明后定义
- 先定义后声明

变量只能定义一次

`extern` 和 `static` 不能连用。

```
int x = 1;
extern int x, y; //无意义
static int y = 2;

const int z = 1;
extern const int z; //无意义

int m, n;
extern int m;
```





2. 变量定义 (3)

变量定义

extern 和 **static** 不能连用。

- 在一个模块内部，一般只定义变量（不声明）。
既定义又声明，没有什么意义。
- 只有对于 **const [static]** 变量，先声明后定义，才有意义（先定义后声明 \Leftrightarrow 只定义不声明），
表明将该变量的作用域修改为整个程序。





3. 变量声明 (1)

变量声明 extern

- 不给变量分配内存。用于声明可能有这个变量（不管这个变量实际上是否存在）。
- 编译器遇到语句 **extern 变量** 时，如果程序并没有访问这个变量，则不做任何处理。否则，进行下面的处理。
- 先在当前模块查找该变量的定义；若没有找到则在其他模块中查找该变量的定义，**若找到且其作用域是整个程序，则成功（否则报错）。**





3. 变量声明 (2)

1.cpp

```
int k = 0;  
const int x = 1;  
extern const int x; //no sense  
extern const int y; //OK  
const int y = 2;  
extern const int z; //OK  
static const int z = 3;  
extern int m; //no sense  
static int m = 3;
```

2.cpp

```
extern const int x;  
extern const int y;  
extern const int z;  
extern int m;  
extern int k = 1; //错, k已经定义  
int main( ) {  
    return x + y + z + m;  
    //找不到 x 和 m  
}
```



深入理解 static, const, extern



華中科技大學

The end.

