

类内数据成员的缺省值



华中科技大学

类内数据成员的缺省值

华中科技大学 计算机学院

金良海 2021.09





类内数据成员的缺省值 (1)

- 在定义一个类时，一般可以设定类中数据成员的缺省值。但对于 static 数据成员，一般不能直接设定缺省值，一个例外的情况是“static const 整型变量”。也就是说，可以在类内定义“static const 整型变量=整数”。
- 整型变量包括：bool、char、short、int、long、long long 及它们的 unsigned 型。



类内数据成员的缺省值 (2)

- 对于类内的“**static const 整型变量**”，一旦在类内设定了缺省值，这个静态变量也就创建了；如果没有在类内设定缺省值，也没有在类外初始化，那么这个静态变量是不存在的。
- 对于类内的**非static变量**，即使设定了缺省值，也能重新赋值（也可以在构造函数的初始化参数列表中重新初始化）。
- 类内数据成员，只能使用等于号(=)设定缺省值，不能直接使用圆括号的方式，即类内只能使用 `int x=1`，而不能使用 `int x(1)`。





类内数据成员的缺省值 (3)

```
class A { /**... **/};
```

```
class B {
```

```
    const A o = A();
```

```
    int i = 0;
```

```
    int &j = i;
```

```
    int num[10] = { 1, 2, 3 };
```

```
    int *const p = num;
```

```
    const int x = 1;
```

```
    volatile const float y = 1.0f;
```

```
    static int a; //static非const的数据成员不能设定初始值
```

```
    static volatile int b; //static非const的数据成员不能设定缺省值
```

```
    static const int c = 2; //static const 的整型变量可以设定缺省值
```

```
    static const int d; //可以设定d的缺省值
```

```
    static const float PI; //static const 的非整型变量不能设定缺省值
```

```
    static const int e[2] = { 1, 2 }; //错, 静态const数组不能设定缺省值
```

```
    int y(1); //错, 只能 int y = 1;
```

```
};
```

```
int B::a = 1;
```

```
volatile int B::b = 1;
```

```
const float B::PI = 3.14159f;
```

```
int main() {
```

```
    cout << B::d; //错, B::d 不存在
```

```
    cout << B::b << B::c; //对
```

```
}
```



类内数据成员的缺省值



华中科技大学

The end.

