c++ explicit 关键字

Tianql 关注

0.564 2017.04.28 14:03:20 字数 395 阅读 15,049

explicit 的作用是用来声明类构造函数是显示调用的，而非隐式调用，所以只用于修饰单参构造函数。因为无参构造函数和多参构造函数本身就是显示调用的。再加上 explicit 关键字也没有什么意义。

explicit 关键字的官方解释：

> This keyword is a declaration specifier that can only be applied to in-class constructor declarations . An explicit constructor cannot take part in implicit conversions. It can only be used to explicitly construct an object 。

从上面的解释可以看出 explicit 的出要作用是：

> - 只能用来修饰类构造函数

- explicit 修饰的构造函数不能被隐式调用
- 禁止类对象之间的隐式转换

举例如下：

```
#include<cstring>
#include<string>
#include<iostream>

class Explicit
{
    private:

    public:
        Explicit(int size)
        {
            std::cout << " the size is " << size << std::endl;
        }
        Explicit(const char* str)
        {
            std::string _str = str;
            std::cout << " the str is " << _str << std::endl;
        }

        Explicit(const Explicit& ins)
        {
            std::cout << " The Explicit is ins" << std::endl;
        }

        Explicit(int a,int b)
        {
            std::cout << " the a is " << a  << " the b is " << b << std::endl;
        }
};

int main()
```

```
{
    Explicit test0(15);
    Explicit test1 = 10;// 隐式调用Explicit(int size)

    Explicit test2("RIGHTRIGHT");
    Explicit test3 = "BUGBUGBUG";// 隐式调用Explicit(const char* str)

    Explicit test4(1, 10);
    Explicit test5 = test1;
}
```

上面的程序虽然没有错误，但是对于 `Explicit test1 = 10;` 和 `Explicit test2 = "BUGBUGBUG";` 这样的句子，把一个 `int` 类型或者 `const char*` 类型的变量赋值给 `Explicit` 类型的变量看起来总归不是很好，并且当程序很大的时候出错之后也不容易排查。所以为了禁止上面那种隐式转换可能带来的风险，一般都把类的单参构造函数声明的显示调用的，就是在构造函数加关键字``explicit`。如下：

```
#include<cstring>
#include<string>
#include<iostream>

class Explicit
{
    private:

    public:
        explicit Explicit(int size)
        {
            std::cout << " the size is " << size << std::endl;
        }
        explicit Explicit(const char* str)
        {
            std::string _str = str;
            std::cout << " the str is " << _str << std::endl;
        }

        Explicit(const Explicit& ins)
        {
            std::cout << " The Explicit is ins" << std::endl;
        }

        Explicit(int a,int b)
        {
            std::cout << " the a is " << a  << " the b is " << b << std::endl;
        }
};

int main()
{
    Explicit test0(15);
    Explicit test1 = 10;// 无法调用

    Explicit test2("RIGHTRIGHT");
    Explicit test3 = "BUGBUGBUG"; // 无法调用

    Explicit test4(1, 10);
    Explicit test5 = test0;
}
```

上面再写 `Explicit test1=10;` `Explicit test3 = "BUGBUGBUG";` 这样的句子的时候程序就会报如下错误：

> error: conversion from ‘int’ to non-scalar type ‘Explicit’ requested
>
> error: conversion from ‘const char [10]’ to non-scalar type ‘Explicit’ requested