

c++ std::conditional详解

std::conditional C++11引入的[类模板](#)，表示的是一种编译期的分支逻辑，我们来看看它的源码吧，其实非常的简单。

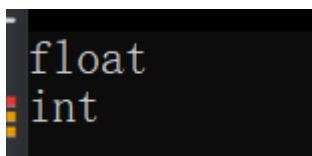
```
// STRUCT TEMPLATE conditional
template <bool _Test, class _Ty1, class _Ty2>
struct conditional { // Choose _Ty1 if _Test is true, and _Ty2 otherwise
    using type = _Ty1;
};

template <class _Ty1, class _Ty2>
struct conditional<false, _Ty1, _Ty2> {
    using type = _Ty2;
};

template <bool _Test, class _Ty1, class _Ty2>
using conditional_t = typename conditional<_Test, _Ty1, _Ty2>::type;
```

从以上可以看到，std::conditional 包含一个[泛化](#)和特化的版本，在结构体里面用type来表示类型模板参数的类型，当第一个非类型模板参数的值为true时，type的类型为第一个类型模板参数的类型，为false时为第二个类型模板参数的值。

```
#include<iostream>
#include <string>
using namespace std;
int main(void)
{
    std::conditional<false, int, float>::type var;
    std::cout << typeid(decltype(var)).name() << std::endl;
    std::conditional<true, int, float>::type var1;
    std::cout << typeid(decltype(var1)).name() << std::endl;
}
```

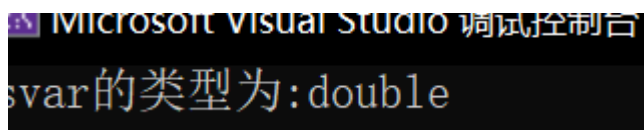


再来看一个复杂点的例子，逻辑上类似与if-else

```

#include<iostream>
#include <string>
using namespace std;
int main(void)
{
    constexpr int j = 135; //假设给进去35
    std::conditional<
        (j > 100), double, //值>100, var是double,
        std::conditional<
            (j > 80), float, //值在80-100之间, 那么var是float类型。
            std::conditional<
                (j > 40), int, //值在40-80之间, 那么var是int类型。
                char //值不超过40, var是char类型
            >::type
        > ::type
    >::type var;
    cout << "tsvar的类型为:" << typeid(decltype(var)).name() << endl;
}

```



Microsoft Visual Studio 调试控制台

tsvar的类型为:double

不过这里的变量j好像只能是常量类型。