

模板类什么时候实现(实例化)

模板实例化

1. 显式实例化:
 - 开发人员明确的告诉模板需要什么样的类型去生成函数或类
2. 隐式实例化:
 - 在编译的过程中由编译器来决定使用什么类型来实例化一个模板

模板具体化

当模板使用某种类型实例化后生成的类或函数不能满足需要时，可以考虑对模板进行具体化。具体化时可以修改原模板的定义，当使用该类型时，按照具体化后的定义实现

```
#include <iostream>
using namespace std;

// #1 模板定义
template<class T>
struct TemplateStruct
{
    TemplateStruct()
    {
        cout << sizeof(T) << endl;
    }
};

// #2 模板显示实例化
template struct TemplateStruct<int>;

// #3 模板具体化
template<> struct TemplateStruct<double>
{
    TemplateStruct() {
        cout << "--8--" << endl;
    }
};

int main()
{
    TemplateStruct<int> intStruct;
    TemplateStruct<double> doubleStruct;

    // #4 模板隐式实例化
    TemplateStruct<char> llStruct;
}
// 输出
// 4
```

// --8--

// 1