单例模式

一个类只能创建一个对象，并且提供一个访问他的全局访问点，该实例被所有程序模块共享。比如在服务器程序中，该服务器的配置信息存放在一个文件中，这些配置数据由一个单例对象统一读取，然后服务进程中的其他对象再通过这个单例对象获取这些配置信息，这种方式简化了在复杂环境下的配置管理。

- **饿汉模式**

```
//饿汉模式

#include<iostream>
class Singleton
{
    private:
    static Singleton _eton;
    Singleton():_data(99){
        std::cout << "单例对象已经构造" << std::endl;
    }
    Singleton(const Singleton& e) = delete;
    ~Singleton(){}
    private:
    int _data;
    public:
    static Singleton & getInstance()
    {
        return _eton;
    }
    int getData()
    {
        return _data;
    }
};
Singleton Singleton:: _eton;
int main()
{
    std::cout << Singleton::getInstance().getData() << std::endl;
    return 0;
}
```