模板

类模板

```cpp
#include <iostream>
template<typename T>

class complex
{
public:
    complex(T r,T a) : re(r) , im(a)
    {}
    T real() const {return re;}
private:
    T re,im;
};

int main()
{
    complex<double> a(1.11,2.0);
    complex<int>     c(5,7);
    std :: cout << a.real() << std::endl;
    std :: cout << c.real() << std::endl;
    return 0;
}
```

函数模板

```cpp
#include <iostream>
template<class T> template<typename T>

inline const T & min(const T &a, const T & b)
{
    return a < b ? b : a;//这里又会操作符重载
}
class stoned
{
public:
    stoned()
    {}
    {}
    stoned(int a, int b): w(a), h(b)
    {}
    bool operator < (const stoned & other) const
    {
        return this->w < other.w;
    }
    ~stoned()
    {}
private:
    int w,h;
};

int main()
{
    stoned r1(1,2);
    stoned r2(2,3);
    stoned r3;
    r3 = min(r1,r2); //这里会根据参数类型自动匹配模板类型，这里不用像类模板一样指定出来具体的模
    return 0;
}
```