# 类模板

建立一个通用类，类中成员的数据类型可以不具体指定

类模板基本语法：

类模板声明：template<class T> class 类名 {... };

类模板实例化：类名<类型> 对象名;

类模板与函数模板的区别：

类模板没有自动类型推导调用模式

类模板在模板参数列表中可以有默认参数

示例：diff.cpp

类模板中成员函数在被调用时才被创建

类模板对象做函数参数，三种方式：

1. 指定传入类型 --直接显示对象的数据类型
2. 参数模板化 --将对象中参数变为模板进行传递
3. 整个类模板化 --将这个对象类型模板化进行传递

示例：parameter.cpp

继承时，当父类是一个模板时，子类在声明时需要指出父类的T的类型

如果想灵活指出父类T类型，子类也要成为模板

模板函数类外实现

template<class T1, class T2>//声明是个模板

Person<T1, T2>::Person(T1 n, T2 a) {

}//Person作用域下

示例：class\_out.cpp

分文件编写，将类模板声明写在.h文件，实现写在.cpp文件

若包含.h文件将报错

原因：类模板函数的创建只有在被调用时创建

解决方法

1. 直接包含.cpp文件
2. 将声明和实现写在同一个文件，后缀.hpp