## 类模板简介

在现今的C++标准模板库中，几乎所有东西都被设计为template形式，不支持模板，就无法使用标准程序库。模板可以认为是针对一个或多个尚未明确的类型而编写一套函数或类型。模板是C++的一个新特性。通过使用模板，C++允许推迟对某些类型的选择，直到想使用模板或者对模板进行专门化处理时。使用模板，可以让程序员面对相似而又略有不同的特性时，更快捷地编写代码，提高开发效率。

本章主要讲述C/C++语言中常用的一些基本概念，如何定义类模板，介绍成员模板、友元模板、函数模板、类模板的参数，最后介绍模板库。

### C++为什么需要模板功能

理解了这个话题，程序员才能深度理解STL，才能用好STL,所以本书的开篇，让我们深入浅出地理解C++引入模板功能的思想。

#### 计算机实在是太傻了

在大多数人的眼里，计算机既神秘又能干。

但在程序员的眼里，计算机实在是又蠢又笨，只不过运算的比人类速度快（电子的速度吗），记忆力好（能存储很多数据）。而且不给指令还什么都干不了，就是给指令，写一个程序，计算机也不灵活。比如，在C++中，同样一个加法，居然要给出不同的数据类型的运算方式。

整数型，我们写这样一个函数：

int add (int m, int n)

{

return m+n;

}

浮点型，我们写这样一个函数：

float add(float i, float j)

{

return i+j;

}

........

每一种类型，我们都必须给计算机写一个函数。

感觉实在是太烦琐了，可是，计算机就是不灵活，它们不具备人类的基本思维。

所以，我们常说电子计算机只是外延和加强了人类的某些功能。

如果说电子计算机能代替人类统治世界，说这话的人实在是太外行了。

#### 1.1.2类和函数重载部分解决了这个问题

可是计算机程序，就是最大限度来模拟人类的思维，让计算机不断接近人类的认知能力，成为人类最好的外延工具之一。所以，远比计算机聪明的科学家和工程师想了很多办法，其中一个是面向对象开发技术。

通过类的封装和函数重载，部分解决了这个问题，比如我们常常这样做。

class calc

{

.....

int add(int m, int n)

{

return m+n;

}

float add(float i, float j)

{

return i+j;

}

......

}