一．创建STL的意图：

为了提升复用性，减少重复的劳动。建立数据结构和算法的一套标准，并且降低他们之间的耦合关系，以提升各自的独立性、弹性、交互操作性。

二．STL基本概念

STL(Standard Template Library,标准模板库)，是惠普实验室开发的一系列软件的统  
称。现在主要出现在 c++中，但是在引入 c++之前该技术已经存在很长时间了。  
STL从广义上分为：容器(container) 算法(algorithm) 迭代器(iterator),容器和算法之间通过迭代器进行无缝连接。

STL 几乎所有的代码都采用了模板类或者模板函数，这相比传统的由函数和类组成的库来说提供了更好的代码重用机会。

三．STL的六大组件

STL提供了六大组件，这些组件之间可以组合套用。这六大组件分别是:容器、算法、迭代器、仿函数、适配器（配接器）、空间配置器。

它们之间的关系： 容器通过空间配置器获得存储的空间，算法通过迭代器存储容器中的内容，仿函数帮助算法完成不同的策略，适配器可以修饰仿函数。

1. 容器

顾名思义，用来存放数据的各种数据结构。如vector，list，deque，set，map。从实现的角度来看，容器是class template。

1. 算法

常用的算法，如sort，find，copy，for\_each。从实现的角度看，STL算法是function template。

1. 迭代器

迭代器是容器与算法之间的桥梁，算法通过迭代器来存储容器中的内容。从实现的角度看，迭代器是重载了多种指针操作的class template。所有容器都有自己专属的迭代器。同时，原生指针也是一种迭代器。

1. 仿函数

行为类似函数，是算法的某种策略。从实现来看，仿函数是一种重载了operator()的class 或者class template。

1. 适配器

一种用来修饰容器或者仿函数或迭代器接口的东西。

1. 空间配置器

负责空间的配置与管理。从实现角度看，配置器是一个实现了动态空间配置、空间管理、空间释放的class tempalte.

四．STL的优点

（一）数据与操作分离

数据是由容器来管理的，而操作是由算法进行管理的。两者之间是独立但又是有联系的，迭代器是两者之间的桥梁。

（二）可重用性

STL中几乎所有代码都是用了模板类和模板函数的形式实现，这样，代码重用性大大提高了。

（三）高性能

速度快，如map是采用红黑树的变体实现的。

（四）高移植性

可以直接移植用STL编写的模块。