**1.STL ATL (c++模板库)**

STL: Standard Template Library，标准模板库(静态库)，惠普实验室开发的一系列软件的统称。它是由Alexander Stepanov、Meng Lee和David R Musser在惠普实验室工作时所开发出来的。从根本上说，STL是一些“容器”的集合，这些“容器”有list,vector,set,map等，STL也是算法和其他一些组件的集合。这里的“容器”和算法的集合指的是世界上很多聪明人很多年的杰作。STL的目的是标准化组件，这样就不用重新开发，可以使用现成的组件。STL现在是C++的一部分，因此不用额外安装什么。

STL可分为容器(containers)、迭代器(iterators)、空间配置器(allocator)、配接器(adapters)、算法(algorithms)、仿函数(functors)六个部分。

ATL: Active Template Library活动模板库(动态库)，是一种微软程序库，支持利用C++语言编写ASP代码以及其它ActiveX程序。通过活动模板库，可以建立COM组件，然后通过ASP页面中的脚本对COM对象进行调用。这种COM组件可以包含属性页、对话框等控件

**2.静态库，动态库，dll,lib**

首先介绍一下静态库（静态链接库）、动态库（动态链接库）的概念，首先两者都是代码共享的方式。

**静态库**：在链接步骤中，连接器将从库文件取得所需的代码，复制到生成的可执行文件中，这种库称为静态库，其特点是可执行文件中包含了库代码的一份完整拷贝；缺点就是被多次使用就会有多份冗余拷贝。即静态库中的指令都全部被直接包含在最终生成的 EXE 文件中了。在vs中新建生成静态库的工程，编译生成成功后，只产生一个.lib文件

**动态库**：动态链接库是一个包含可由多个程序同时使用的代码和数据的库，DLL不是可执行文件。动态链接提供了一种方法，使进程可以调用不属于其可执行代码的函数。函数的可执行代码位于一个 DLL 中，该 DLL 包含一个或多个已被编译、链接并与使用它们的进程分开存储的函数。在vs中新建生成动态库的工程，编译成功后，产生一个.lib文件和一个.dll文件

那么上述静态库和动态库中的lib有什么区别呢？

**静态库中的lib**：该LIB包含函数代码本身（即包括函数的索引，也包括实现），在编译时直接将代码加入程序当中

**动态库中的lib**：该LIB包含了函数所在的DLL文件和文件中函数位置的信息（索引,声明），函数实现代码运行由加载在进程空间中的DLL（包含函数的实际信息）提供。

**总之，lib是编译时用到的，dll是运行时用到的。如果要完成源代码的编译，只需要lib；如果要使动态链接的程序运行起来，只需要dll**。

Ps: <https://www.cnblogs.com/ouyangping/p/7941193.html>

**3.MFC**

微软基础类库（[Microsoft](https://baike.baidu.com/item/Microsoft/125917) [Foundation](https://baike.baidu.com/item/Foundation/8927757) Classes，简称MFC）是[微软公司](https://baike.baidu.com/item/%E5%BE%AE%E8%BD%AF%E5%85%AC%E5%8F%B8/732128)提供的一个[类库](https://baike.baidu.com/item/%E7%B1%BB%E5%BA%93/3351433)（class libraries），以[C++类](https://baike.baidu.com/item/C%2B%2B%E7%B1%BB/9567879)的形式封装了[Windows API](https://baike.baidu.com/item/Windows%20API/6088382)，并且包含一个应用程序框架。其中包含大量[Windows](https://baike.baidu.com/item/Windows/165458)[句柄](https://baike.baidu.com/item/%E5%8F%A5%E6%9F%84/3527587)封装类和很多Windows的内建[控件](https://baike.baidu.com/item/%E6%8E%A7%E4%BB%B6/3351794)和组件的封装类。

MFC Object和Windows Object的关系:

Windows Object: 是Win32下用句柄表示的Windows操作系统对象;

MFC Object : 是C++对象，是一个C++类的实例，这里MFC Object是有特定含义的，指封装Windows Object的C++ Object，并非指任意的C++ Object.

举例： 窗口：

MFC Object: CWnd and CWnd-derived classes

Windows Object: HWND

PS:详细见百度百科<https://baike.baidu.com/item/MFC/2530850?fr=aladdin>

**4.COM组件(Componet Object Model,组件对象模型)**

我的理解一套com组件就是一个frm加若干个Mod,包含了各种接口，和com类(c++类)。 是一种规范, 连接机制和协议. 一个组件就是一个二进制对象

详细可见: https://www.cnblogs.com/gd-luojialin/p/9346706.html

**5.OLE(Object Linking and Embedding)**

OLE 不仅是[桌面应用程序](https://baike.baidu.com/item/%E6%A1%8C%E9%9D%A2%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F/2331979)集成，而且还定义和实现了一种允许应用程序作为软件“对象”（数据[集合](https://baike.baidu.com/item/%E9%9B%86%E5%90%88/73081)和操作数据的[函数](https://baike.baidu.com/item/%E5%87%BD%E6%95%B0/301912)）彼此进行“连接”的[机制](https://baike.baidu.com/item/%E6%9C%BA%E5%88%B6/1433787)，这种连接机制和协议称为[组件对象模型](https://baike.baidu.com/item/%E7%BB%84%E4%BB%B6%E5%AF%B9%E8%B1%A1%E6%A8%A1%E5%9E%8B/3351546)（COM）. 是一种[面向对象](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1)的技术.

OLE是技术，com是一种对于这种技术的规范

Ps: <https://www.cnblogs.com/olawonzhu/articles/1683424.html>

**6．ActiveX**

[ActiveX](https://baike.baidu.com/item/ActiveX)是Microsoft对于一系列策略性[面向对象程序](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E7%A8%8B%E5%BA%8F/8359543)技术和工具的称呼.

ActiveX控件是用于互联网的很小的程序，有时称为插件程序*。*它们会允许播放动画，或帮助执行任务。在创建包括ActiveX程序时，主要的工作就是组件，一个可以自足的在ActiveX网络（Windows，Mac，Linux）中任意运行的程序。这个组件就是ActiveX控件。ActiveX控件等价于以前的OLE控件或OCX。

ActiveX控件是[Microsoft](https://baike.baidu.com/item/Microsoft)的ActiveX技术的一部分。ActiveX控件是可以在应用程序和网络中计算机上重复使用的程序对象。创建它的主要技术是Microsoft的ActiveX技术，其中主要是组件对象模型（COM）。

**7.CAA调用ActiveX控件**

同样是COM组件，可以相互调用。

示例：EXCEL

1. 导出Excel 类型库中的类
2. 创建MFC 应用程序，点击完成
3. 项目->添加类->MFC->1/4类型（ActiveX控件中的MFC类，或者类型库中的MFC类），找到对应类型库，添加。
4. 整理.h和.cpp文件添加至CAA模块中。
5. 创建一个.h文件，把导出的所有关于excel的头文件封装到一个文件中，同理cpp。
6. 正常使用。

PS：<https://blog.csdn.net/qq_38149046/article/details/80371992> （图片有误）

**8. CAA调用外部dll**

<https://blog.csdn.net/originalcandy/article/details/79788748>

**9.关于宽字符，单字符（w\_char,char），多字节字符，字符集，编码**

**字符集**：这是一个表，记录每个字符对应的数值。有ASCII(单字节)字符集（也是编码方案），UNICODE字符集，GB2312（中文双字节）中文字符集,GBK字符集(中英文双字节，没有特殊的编码方式，一般称呼GBK编码)

**字符编码：**这是字符集中字符对应数值的表现形式. Ps：比如字符 € 的字符集数值是8364（用十六进制表示为U+20AC），而是用UTF-8字符编码的结果就是E2 82 AC。

**Unicode(USC-2)字符集:** 从一些多语言软件制作厂家组织在一起的 Unicode Project 中诞生的，可以容纳全世界所有语言的编码方案.但是他只是规定了符号的二进制代码，却没有规定这些二进制如何存储,所以就出现了UTF-8等编码，是字符集的实现方式（也就是在unicode码基础上，添加另外的规则，使其存储于计算机中）。包含UTF-8，UTF-32等字符集。在COM里面只支持UNICODE

**UTF-8：**不受字节序影响；(中文占3个字节)

1. 128以内的，也就是 0x00-0x7F，兼容 ASCII 编码，均用1个字节表示。

UTF-8的编码规则很简单, 只有两条:

1) 对于单字节的符号, 字节的第一位设为0, 后面7位为这个符号的unicode码. 因此对于英语字母, UTF-8编码和ASCII码是相同的.

2) 对于n字节的符号(n>1), 第一个字节的前n位都设为1, 第n+1位设为0, 后面字节的前两位一律设为10. 剩下的没有提及的二进制位, 全部为这个符号的unicode码.

**wchar\_t ,char**:这是程序运行时字符存储的地方，因此为内部表现形式，而编码则是外部表现形式。

**ASCII码：**美国制定了一套字符编码, 对英语字符与二进制位之间的关系, 做了统  
一规定. 这被称为ASCII码，一共规定了128个字符的编码。但是只能适用于美国，其他国家有其自己不同的字母，所以需要扩充,**但是所有编码的0-127位都是一样的**.(中文常见编码GB2312,GBK是其扩展包含了繁体字, Big5主要用于台湾香港等繁体字，这三种是常见的中文编码)

**ANSI**: (256字符)其实ANSI并不是某一种特定的字符编码，而是在不同的系统中，ANSI表示不同的编码。你的美国同事Bob的系统中ANSI编码其实是ASCII编码（ASCII编码不能表示汉字，所以汉字为乱码），而你的系统中（“汉字”正常显示）ANSI编码其实是GBK编码，而韩文系统中（“한국어”正常显示）.ANSI就是在ASCII基础上，每个国家在其后面扩充自己国家的符号。ANSI编码其实是EUC-KR编码。微软用一个叫“[Windows code pages](https://en.wikipedia.org/wiki/Code_page)”（在命令行下执行chcp命令可以查看当前code page的值）的值来判断系统默认编码，比如：简体中文的code page值为936（它表示GBK编码，win95之前表示GB2312，详见：[Microsoft Windows' Code Page 936](https://en.wikipedia.org/wiki/Code_page_1386)），繁体中文的code page值为950（表示Big-5编码）。

**ANSI和Unicode**: 这是两种不同的编码方式标准，ANSI中的字符采用8bit，而UNICODE中的字符采用16bit。（对于字符来说ANSI以单字节存放英文字符，以双字节存放中文等字符,而Unicode下，英文和中文的字符都以双字节存放）Unicode码也是一种国际标准编码，采用二个字节编码，与ANSI码不兼容

**什么是BOM**：带BOM（签名）的UTF-8文本文件与不带BOM的UTF-8文本文件的区别在于：前者在文件的开头有3个字节 EF BB BF.

<https://blog.csdn.net/ice_ly000/article/details/85159459>

https://www.cnblogs.com/codingmengmeng/p/11028744.html

**宽字符（wchar）**：用多个[字节](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82/1096318)来代表的字符称之为宽字符，[Unicode](https://baike.baidu.com/item/Unicode/750500)只是宽[字符编码](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6%E7%BC%96%E7%A0%81)的一种实现，宽字符并不一定是Unicode

**单字符(char) :**占8位,只能表示十进制0-127之间的一个字节

Ansi：

char   代表一个字符  （CHAR）

char \*  代表一个字符串指针   (PCHAR    PSTR   LPSTR)

const char \*   代表常量字符串的指针  (LPCSTR)

● **Unicode：**

wchar\_t   WCHAR

wchar\_t \*     PWCHAR、PWSTR、LPWSTR

赋值时注意：str=L”liming”     **加上L**

const wchar\_t \*    LPCWSTR

● T 通用类型：

**TCHAR**、(**TCHAR \*** 、PTCHAR、PTSTR、LPTSTR)、LPCTSTR

赋值时注意：str=\_T(”liming”)     **加上\_T( )**

\_T( )是一个宏，可以将里面的字符串转化成所在工程环境中的字符类型  
TCHAR是UNICODE的一种，等同于\_T()这个宏

<https://blog.csdn.net/tge7618291/article/details/7599902>

<https://blog.csdn.net/wyansai/article/details/50764315>(转化)

<https://blog.csdn.net/qq_42068856/article/details/83792174>（字符集，编码详解）

<https://www.pconline.com.cn/pcedu/empolder/gj/other/0505/616631.html>

[https://www.cnblogs.com/wllwqdeai/p/9795123.html 。(ANSI](https://www.cnblogs.com/wllwqdeai/p/9795123.html%20%20%20%20。(ANSI)和Unicode)

<https://blog.csdn.net/imxiangzi/article/details/77370160> (关于ANSI)

<https://www.cnblogs.com/malecrab/p/5300503.html（UTF-8>）

<https://www.cnblogs.com/LWX-YEER/p/11203185.html>（位，字节，字符，编码。。。）

**10.静态成员函数和普通成员函数的区别**

非静态的函数由类对象(加.或指针加->;)调用,这时将向函数传递this指针.而静态函数由类名(::)(或对象名.)调用,但静态函数不传递this指针,不识别对象个体,所以通常用来对类的静态数据成员操作.

且调用静态数据成员时候，不用函数名，直接调用，具体可见链接。

https://blog.csdn.net/aimei9103/article/details/101392665?depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-BlogCommendFromBaidu-1&utm\_source=distribute.pc\_relevant.none-task-blog-BlogCommendFromBaidu-1