C语言强化

C语言实训2020

李巍
Wei Li
四川师范大学
liw@sicnu.edu.cn

准备

项目文档

- 项目文档介绍

获得资源

工欲善其事，必先利其器

Git：https://gitforwindows.org/

Github：https://github.com/

VS：https://visualstudio.microsoft.com/zh-hans/vs/

本课资源

Available：https://github.com/TeacherWLee/sicnu-ctrain-2020

Clone：https://github.com/TeacherWLee/sicnu-ctrain-2020.git

Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

TeacherWLee / sicnu-ctrain-2020

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Actions | Projects 0 | Wiki | Security 0 | Insights | Settings

No description, website, or topics provided.

Edit

Manage topics

5 commits | 1 branch | 0 packages | 0 releases | 1 contributor | GPL-3.0

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

TeacherWLee add vs2019 installer — Latest commit 0f89b11 12 minutes ago

P2PSharer	add P2PSharer for VS2019	13 hours ago
documents	add vs2019 installer	12 minutes ago
sicnu-ctrain-2020	add demo2-chat	7 hours ago
.gitignore	Initial commit	yesterday
LICENSE	Initial commit	yesterday
README.md	Initial commit	yesterday

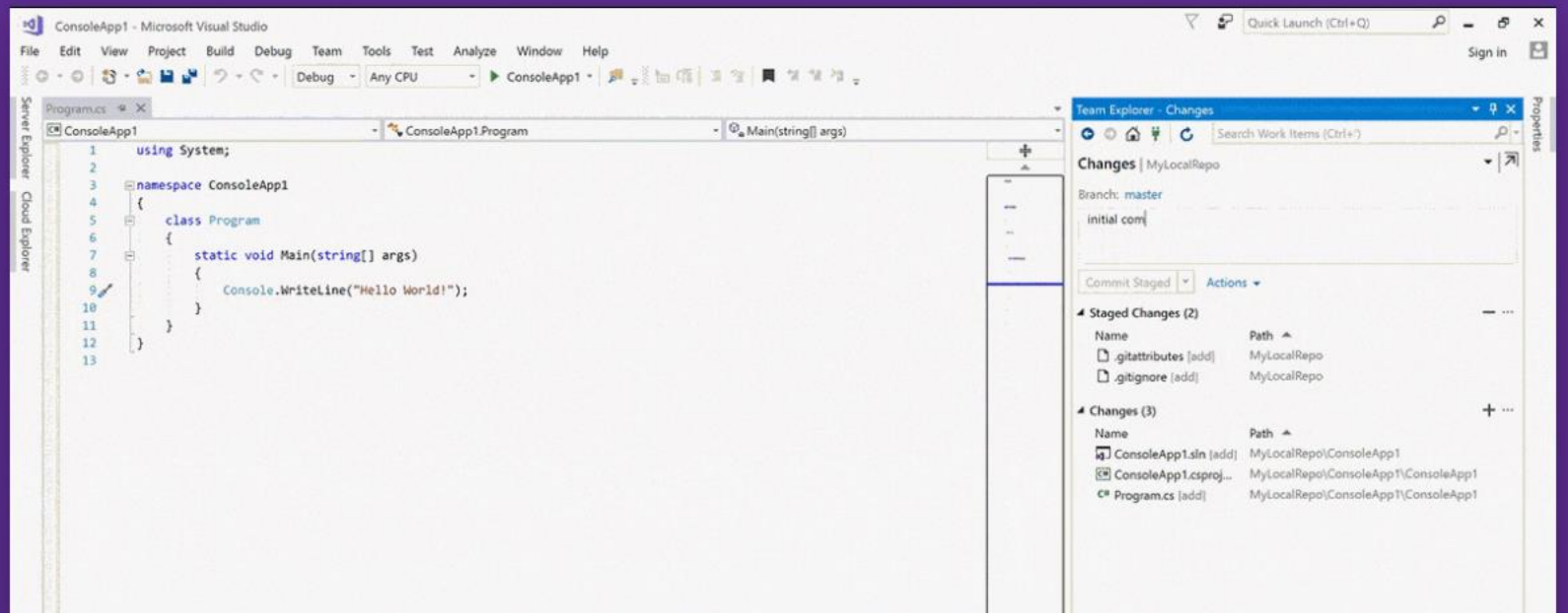
README.md

sicnu-ctrain-2020

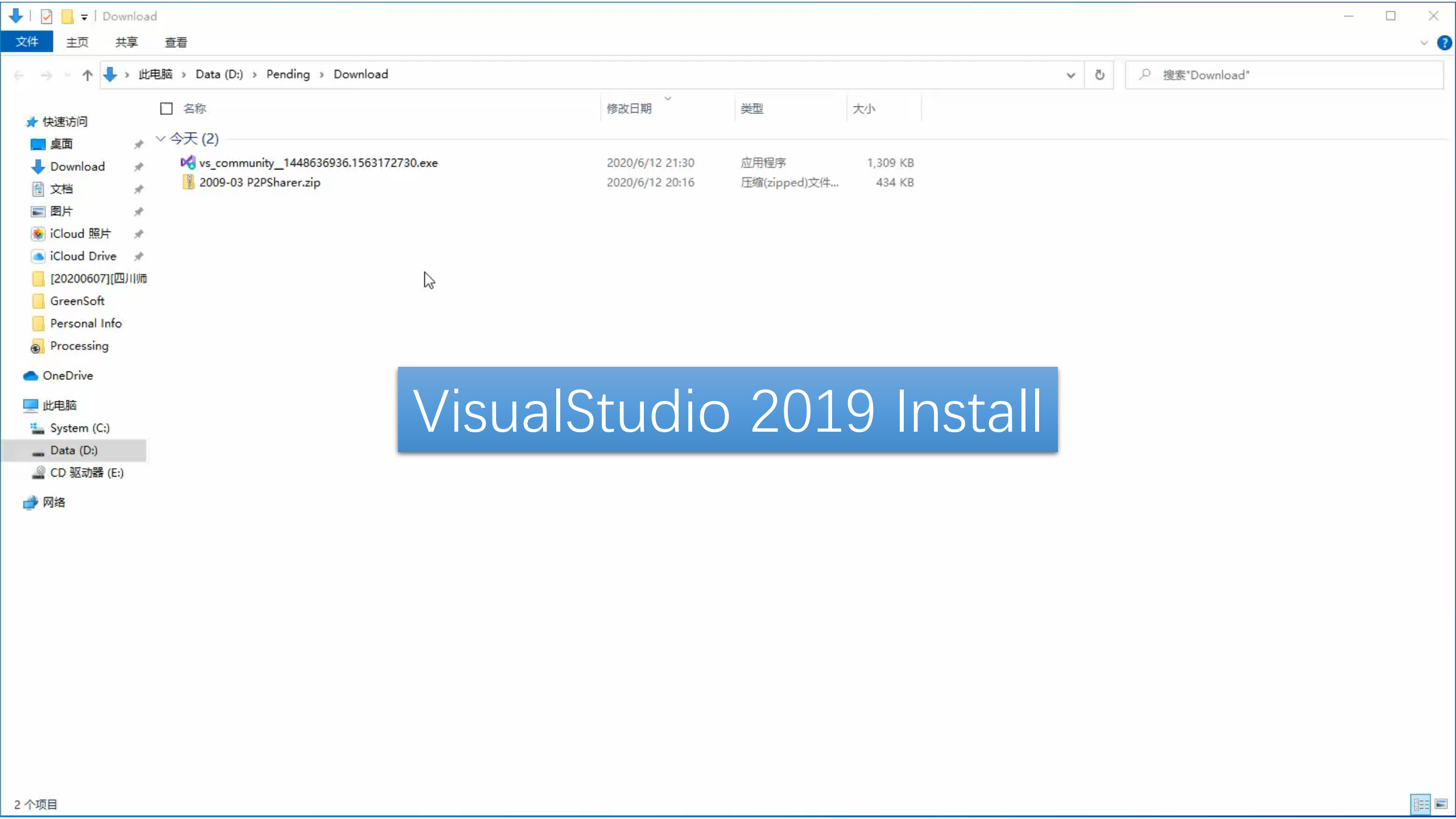
Visual Studio 2019

更快地进行代码编写。 更智能地执行操作。 使用同类最佳 IDE 创建未来。

下载 Visual Studio



https://visualstudio.microsoft.com/zh-hans/vs/

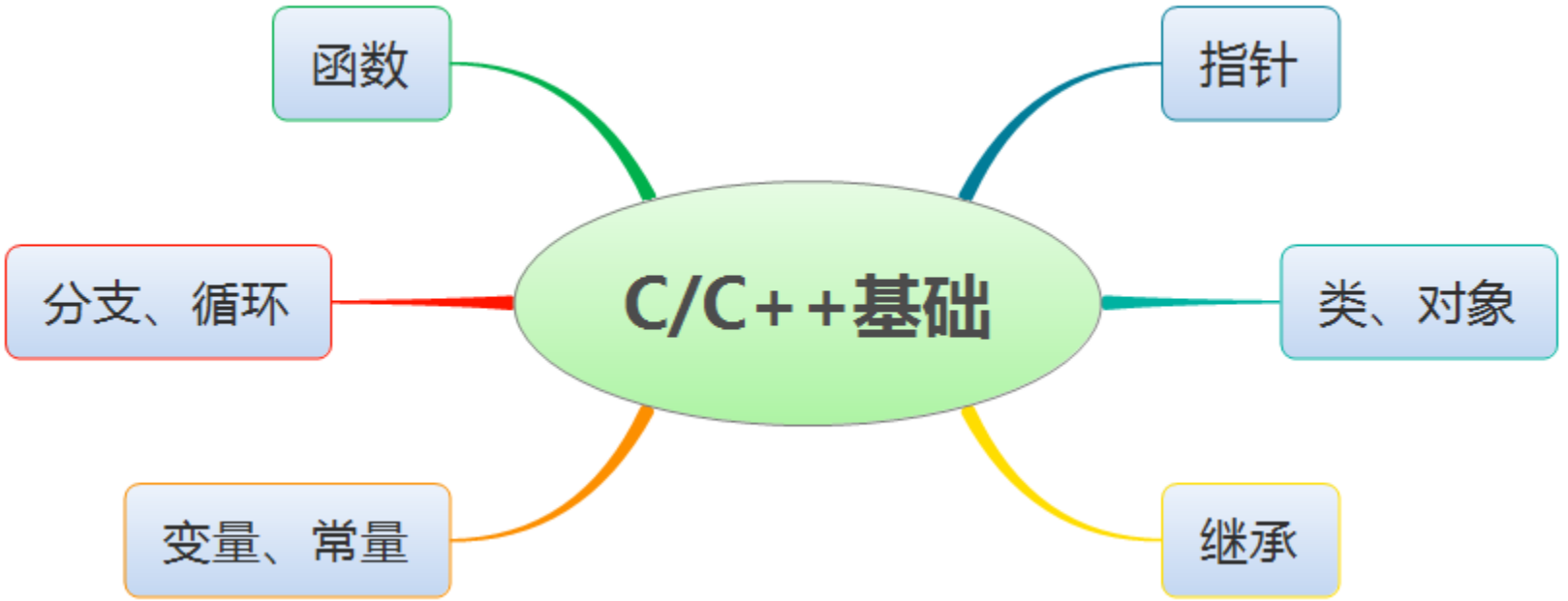

Download
文件
主页
共享
查看
此电脑 › Data (D:) › Pending › Download
搜索"Download"
名称
修改日期
类型
大小
今天 (2)
vs_community__1448636936.1563172730.exe
2020/6/12 21:30
应用程序
1,309 KB
2009-03 P2PSharer.zip
2020/6/12 20:16
压缩(zipped)文件...
434 KB
快速访问
桌面
Download
文档
图片
iCloud 照片
iCloud Drive
[20200607][四川师
GreenSoft
Personal Info
Processing
OneDrive
此电脑
System (C:)
Data (D:)
CD 驱动器 (E:)
网络
VisualStudio 2019 Install
2 个项目

基础

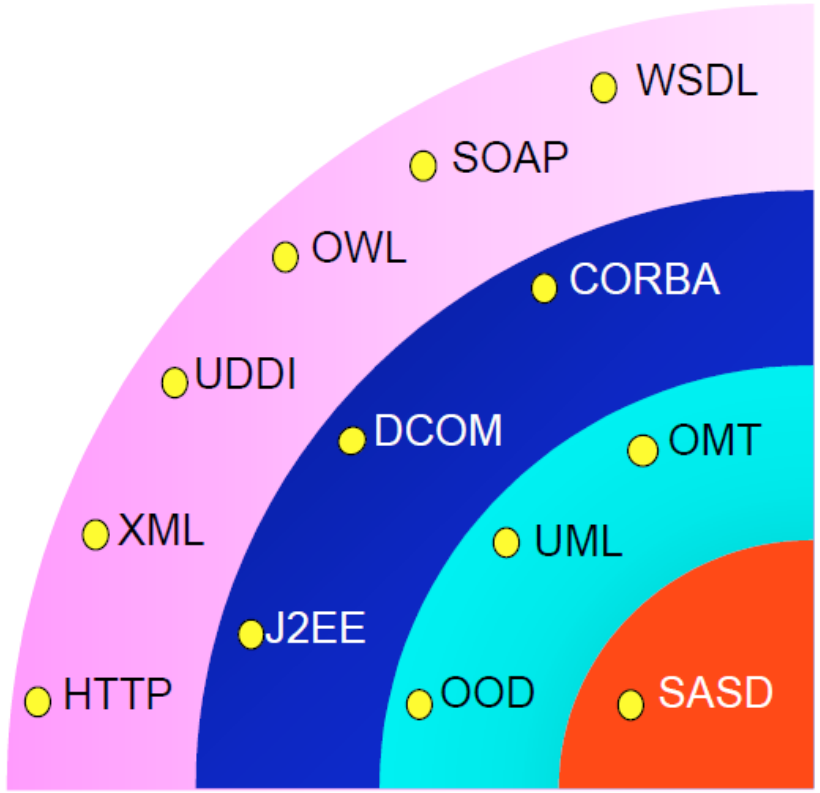
C/C++基础

基础

C/C++基础



工程思维基础



面向服务：在应用表现层次上将软件构件化，即应用业务过程由服务组成，而服务由构件组装而成。

面向构件：寻求比类的粒度更大的且易于复用的构件，期望实现软件的再工程 。

面向对象：以类为基本程序单元，对象是类的实例化，对象之间以消息传递为基本手段。

面向过程：以算法作为基本构造单元，强调自顶向下的功能分解，将功能和数据进行一定程度的分离。

编码规范

课堂讨论：

- 各位同学写的代码是不是处于“只要实现功能，编译通过即可”的状态？
- 当编译器提示警告信息时，你是一定要修正？还是下意识忽略？
- 程序员写的代码是给机器看的？还是给人看的？

编码规范

C++

```
#include "stdafx.h"
          #include          "stdio.h"
        void test();
               int _tmain
               (int argc,
                  _TCHAR*          argv[])
                     { test(); return
                              0; }
                           char C[25][40];void d(int x,int y)
           {C[x][y]=C[x][y+1]=32;}int f(int x){return (int)x*x*.08;}
           void test(){int i,j;                              char s[5]="TEST";
        for(i=0;i<25;i++)
for(j=0;j<40;j++)
       C[i][j]=s[(i+j)%4];
      for(i=1;i<=7;i++)
      {d(18-i,12);
      C[20-f(i)][i+19]=
            C[20-f(i)][20-i]=32;
   }d(10,13);d(9,13);
         d(8,14);d(7,15);
            d(6,16);d(5,18);d(5,20);                    d(5,22);d(5,26);
d(6,23);d(6,25);d(7,25);for(i=0;i<25;i++,printf("\n"))
              for(j=0;j<40;printf("%c",C[i][j++]));}
```

编码规范

编码规范

编码规范


```
def fval(i):
    ret = 2
    n1 = 1
    n2 = 1
    i2 = i - 3
    while i2 >= 0:
        n1 = n2
        n2 = ret
        ret = n2 + n1
        i2 -= 1
    return 1 if i < 2 else ret
```

编码规范

```python
def fval(i):
    ret = 2
    n1 = 1
    n2 = 1
    i2 = i - 3
    while i2 >= 0:
        n1 = n2
        n2 = ret
        ret = n2 + n1
        i2 -= 1
    return 1 if i < 2 else ret
```

```python
def fibonacci(position):
    if position < 2:
        return 1

    previous_but_one = 1
    previous = 1
    result = 2
    for n in range(2, position):
        previous_but_one = previous
        previous = result
        result = previous + previous_but_one
    return result
```

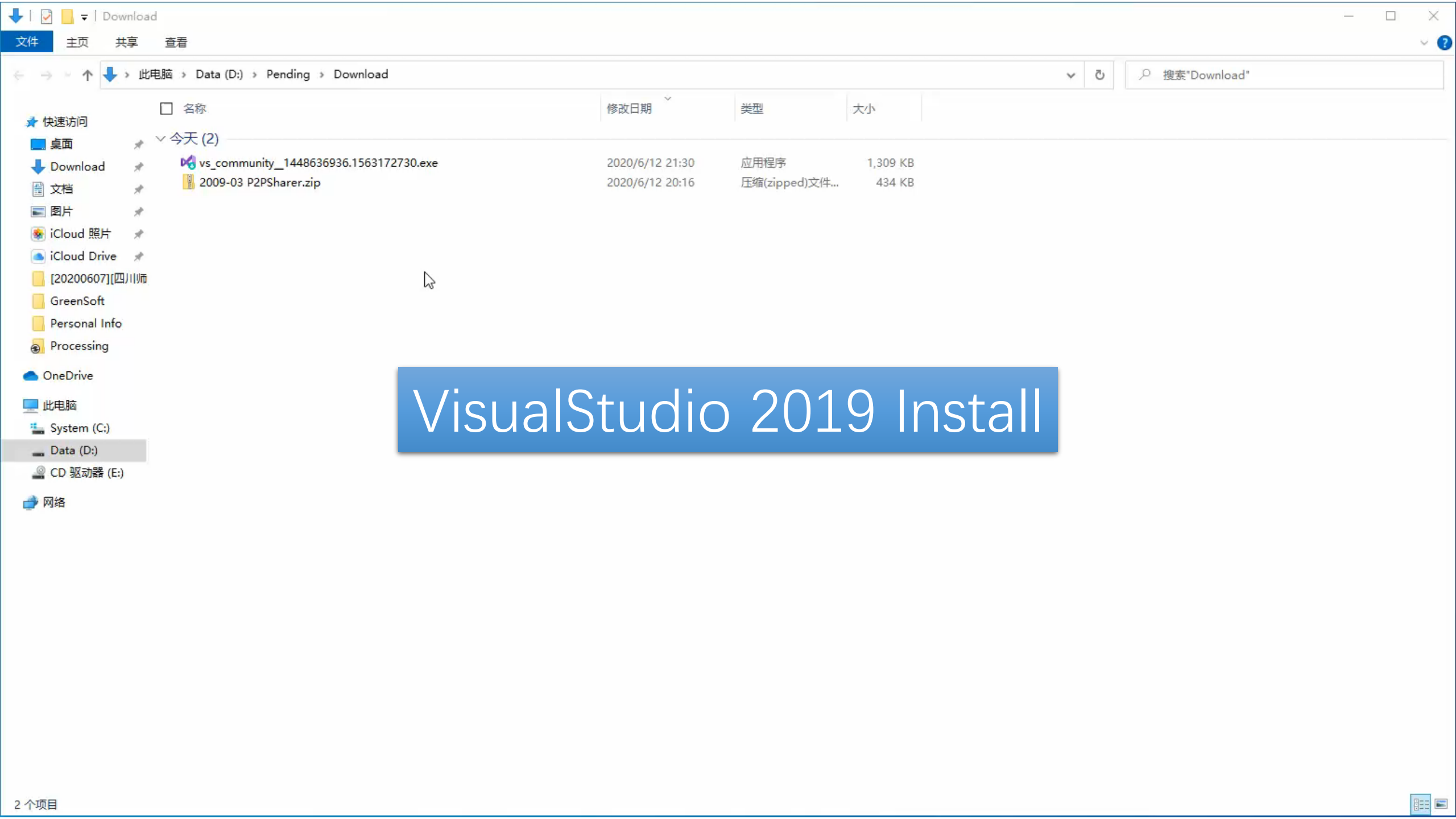
Google编码规范

https://github.com/google/styleguide

Our C++ Style Guide, Objective-C Style Guide, Java Style Guide, Python Style Guide, Shell Style Guide, HTML/CSS Style Guide, JavaScript Style Guide, AngularJS Style Guide, Common Lisp Style Guide, and Vimscript Style Guide are now available. We have also released cpplint, a tool to assist with style guide compliance, and google-c-style.el, an Emacs settings file for Google style.

If your project requires that you create a new XML document format, our XML Document Format Style Guide may be helpful. In addition to actual style rules, it also contains advice on designing your own vs. adapting an existing format, on XML instance document formatting, and on elements vs. attributes.

Visual Studio 2019 Community


Download
文件
主页
共享
查看
此电脑 › Data (D:) › Pending › Download
搜索"Download"
名称
修改日期
类型
大小
今天 (2)
vs_community__1448636936.1563172730.exe
2020/6/12 21:30
应用程序
1,309 KB
2009-03 P2PSharer.zip
2020/6/12 20:16
压缩(zipped)文件...
434 KB
快速访问
桌面
Download
文档
图片
iCloud 照片
iCloud Drive
[20200607][四川师
GreenSoft
Personal Info
Processing
OneDrive
此电脑
System (C:)
Data (D:)
CD 驱动器 (E:)
网络
VisualStudio 2019 Install
2 个项目

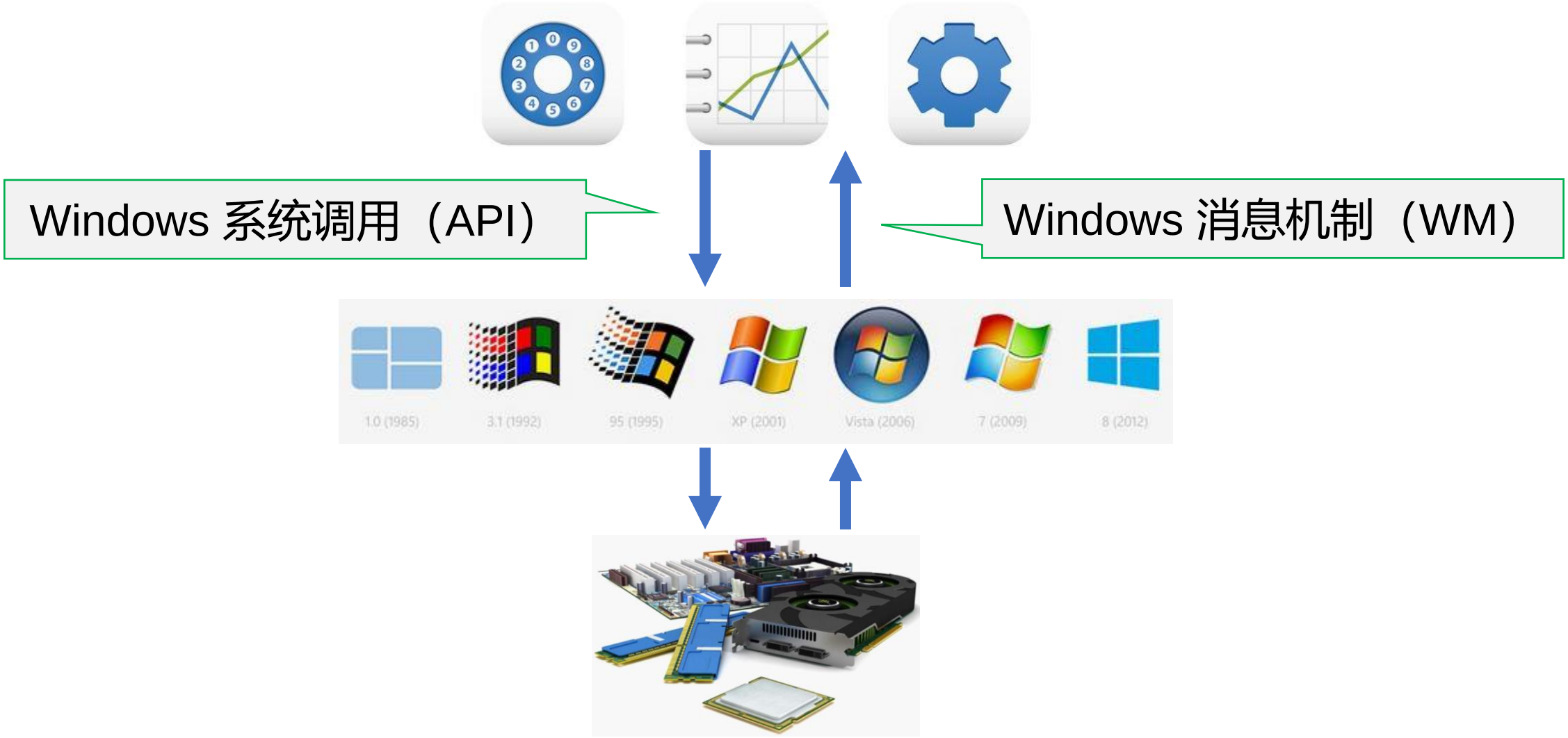
VS2019基础设置

- 基本结构
- 主题
- 字体
- 可定制型

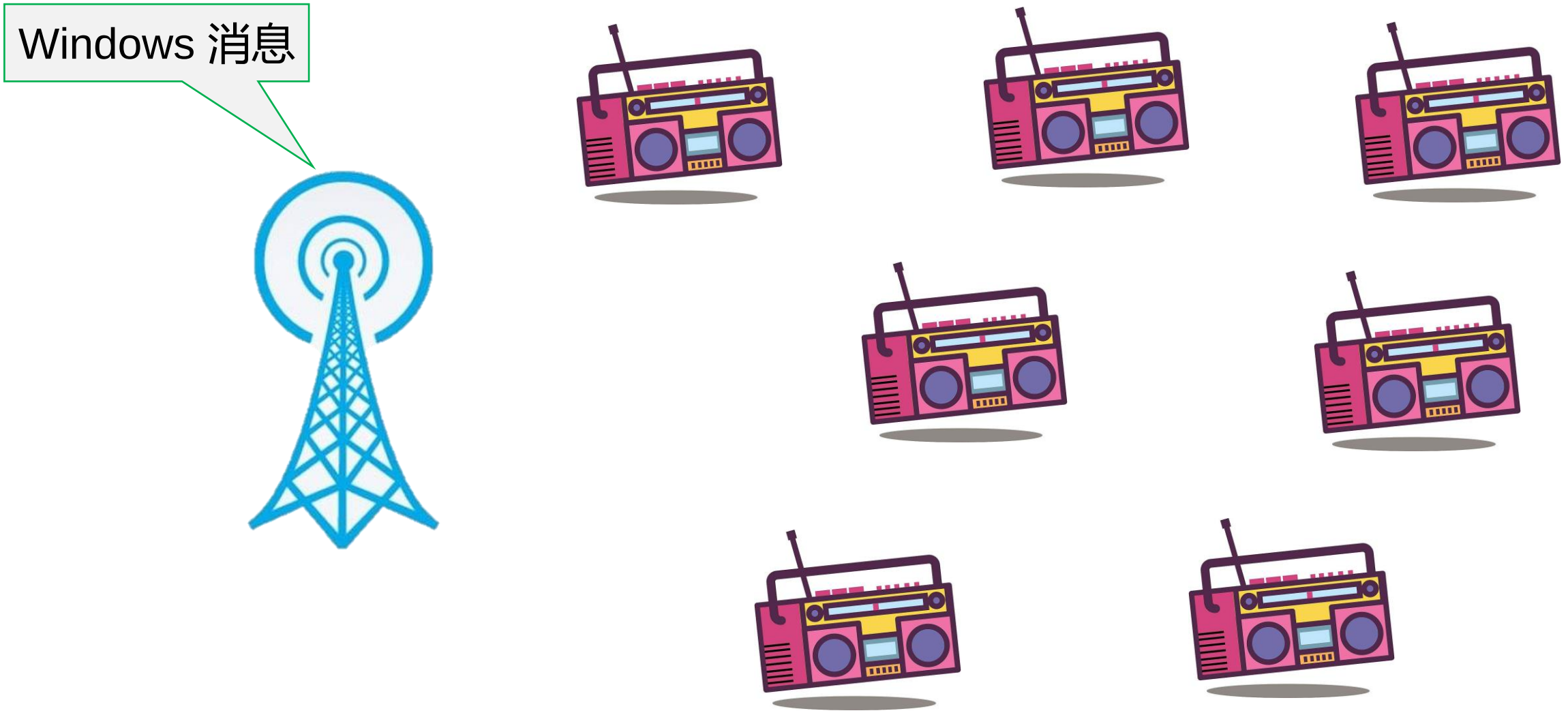
微软基础类库MFC

Microsoft Foundation Class

Windows信息机制



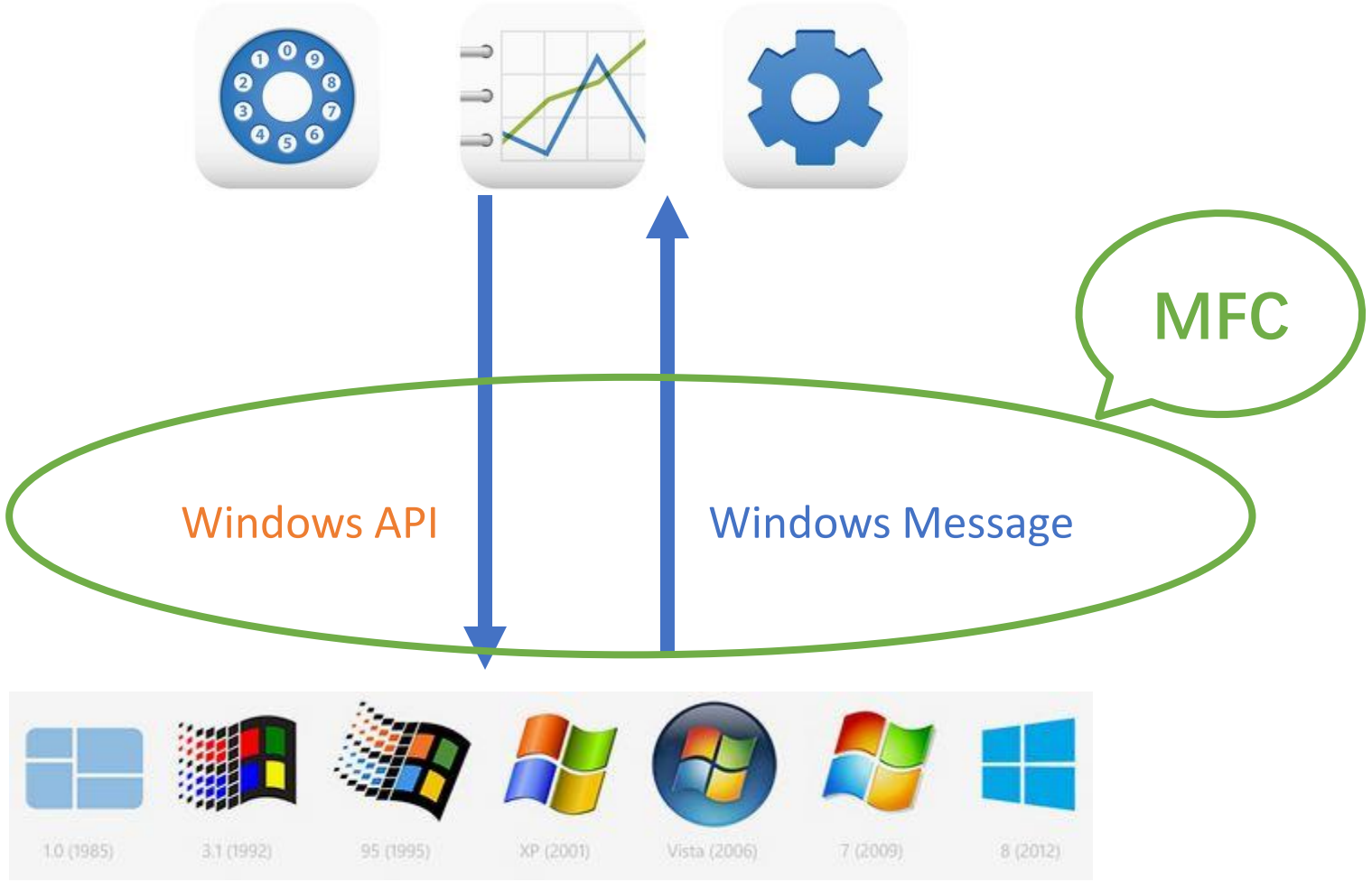
Windows信息机制



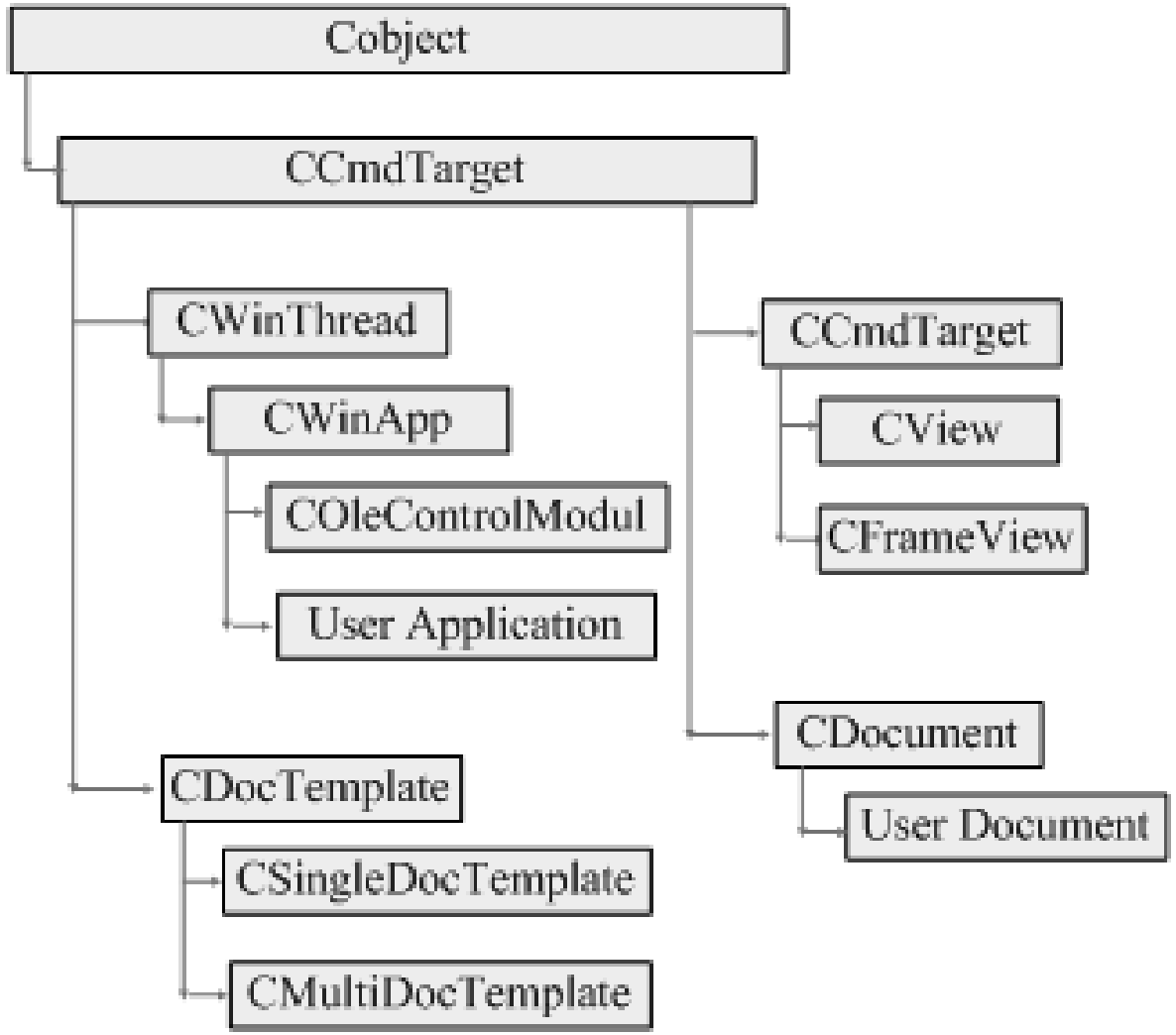
MFC简介

- 微软基础类库（MFC，Microsoft Foundation Class）是微软为Windows程序员提供的一个面向对象的Windows编程接口，以类的层次形式进行组织，高层提供一般功能，低层实现具体行为。低层继承自高层。
- MFC能处理很多与Windows相关的常见任务。可以实现消息循环，提供易学易用的成员函数，比如在onLButtonDown()种插入代码，处理窗口消息
- MFC还提供应用程序开发模型，称为文档/视图模型。此模型将应用程序数据与用户界面元素分离，允许两部分程。
- MFC采用了Windows API中的一些功能，并为程序员提供更友好的C++类，使之更易于使用。
- MFC由很多C++类组成。其中有些类，例如CWnd或CWinThread，是整个框架中大部分内容的基础。这些类封装了基本功能，如大多数Windows应用程序都需要打开窗口功能。还有其他专用类，如CSplitterWnd是从这些类派生来的，继承了父类的所有特性，还增加了自己的功能。开发人员还可以创建自定义类，执行特定任务。序独立存在。

MFC概念


MFC
Windows API
Windows Message
1.0 (1985)
3.1 (1992)
95 (1995)
XP (2001)
Vista (2006)
7 (2009)
8 (2012)

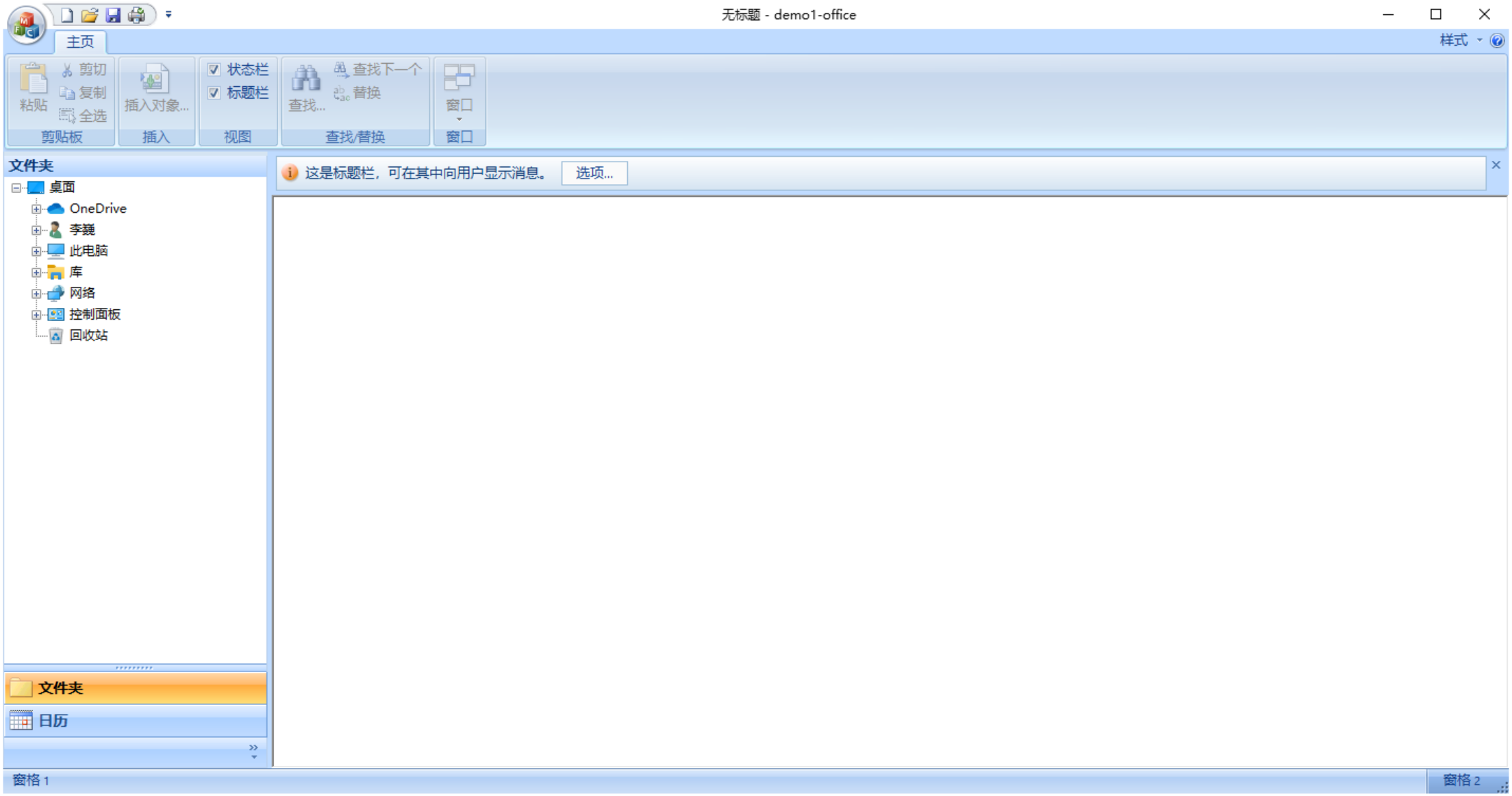
MFC类层次结构图



MFC 类库层次结构

- 入口函数就是指一个程序的入口点。WinMain函数是Windows程序的入口函数。
- 从创建的MFC程序中，并不能找到WinMain函数。这是因为典型Windows程序的大部分初始化工作都是标准化的，因此把WinMain函数隐藏在应用程序的框架中。当一个程序编译时，会自动将该函数链接到程序中。

MFC项目创建



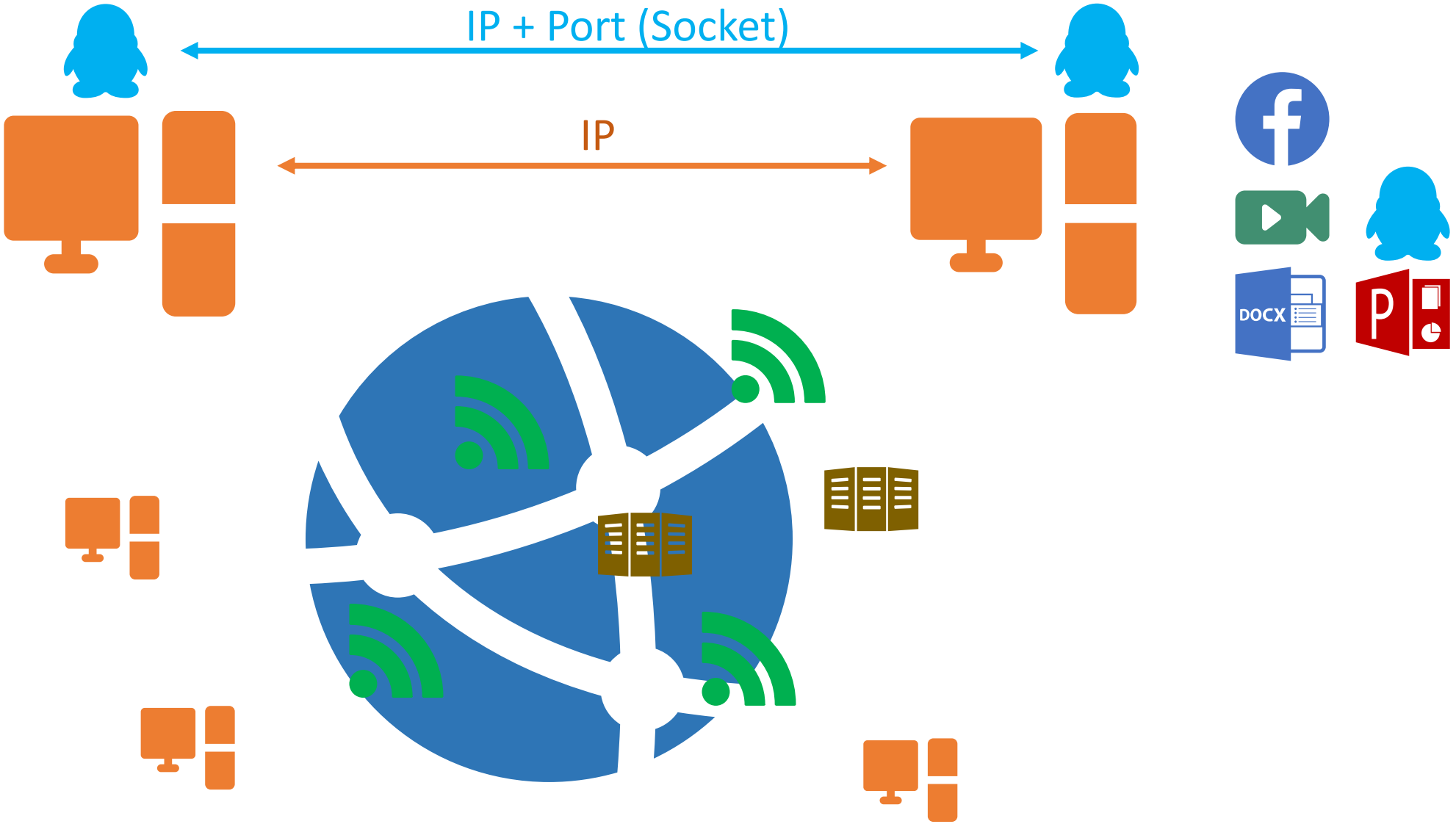
liw@sicnu.edu.cn

套接字Socket

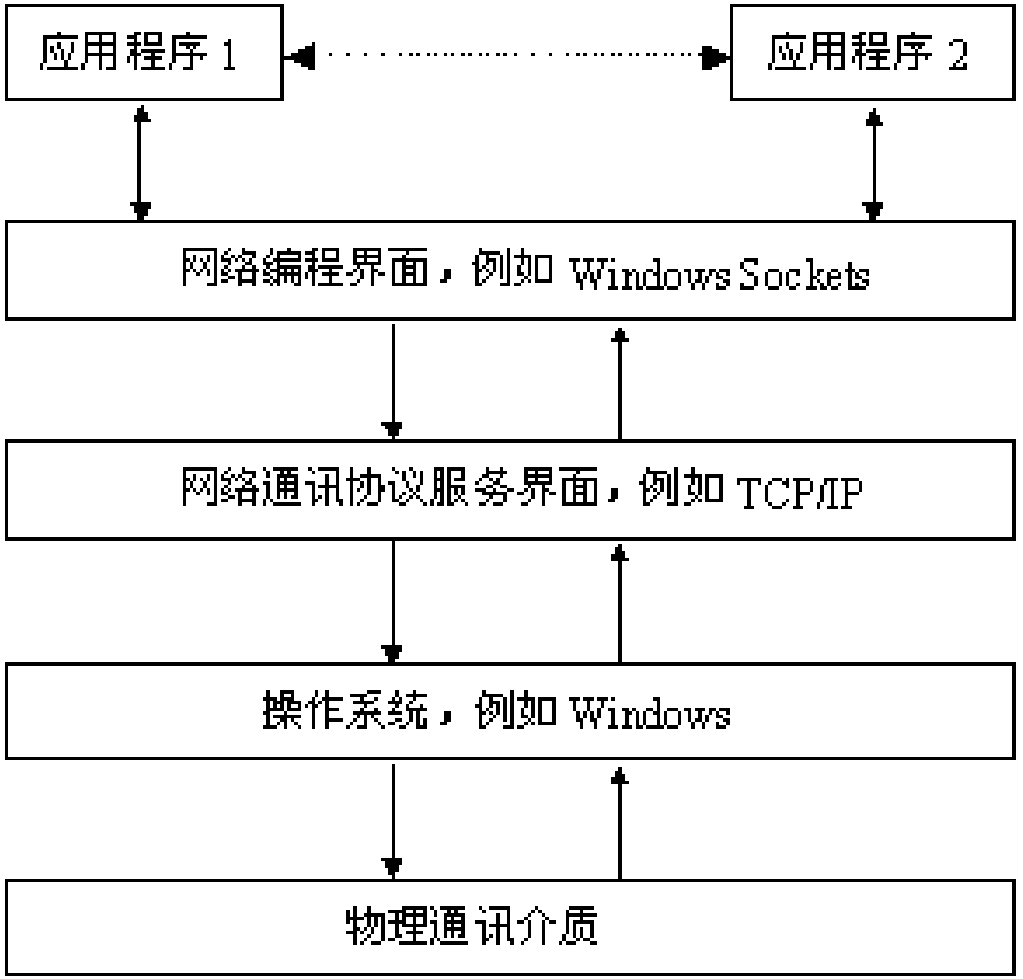
Socket简介

- 在计算机通信领域，socket 被翻译为“套接字”，它是计算机之间进行通信的一种约定或一种方式。通过 socket 这种约定，一台计算机可以接收其他计算机的数据，也可以向其他计算机发送数据。
- socket起源于Unix，而Unix/Linux基本哲学之一就是“一切皆文件”，都可以用“打开open –> 读写write/read –> 关闭close”模式来操作。
- Socket是该模式的一个实现：即socket是一种特殊的文件，一些socket函数就是对其进行的操作（读/写IO、打开、关闭）。
- Socket()函数返回一个整型的Socket描述符，随后的连接建立、数据传输等操作都是通过该Socket实现的。
- Windows应用程序可以有无限的网络功能，都是建立在WinSock接口的基础上。WinSock是Windows Sockets的简称，也称为Windows套接字，是微软根据BSD UNIX操作系统中流行的Berkeley套接字规范而实现的一套Micosoft Windows下的网络编程接口。

Socket简介

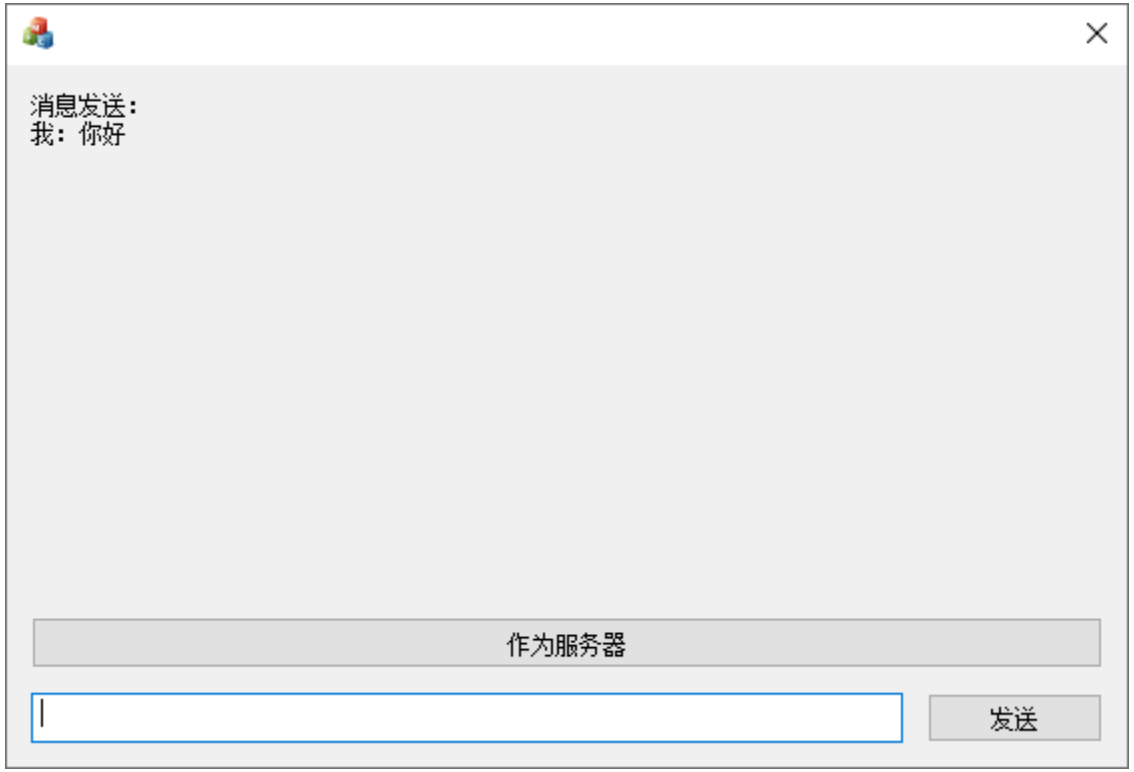


Socket简介



TCP/IP协议核心与应用程序关系

Socket聊天程序



四川師範大學

SICHUAN NORMAL UNIVERSITY