# **为什么选择visual studio 2017**

1. 本机已经安装完开发环境。
2. 开发语言为C++，契合windows系统。

**目前存在的问题**

1、windows桌面c++开发以前从未接触过，需要进行一定的基础桌面开发学习。

2、没有现成的类似系统源码或框架进行参考，开发过程需要自己摸索。

# **Win32应用程序开发：完整的开发流程**

## **一、WinMain入口点**

我们在学习标准C++的时候，都知道每个应用程序运行时都会先进入入口点函数main，而当从main函数跳出时程序就结束了。在Windows编程里面，也是一样的，只是我们的入口点函数不叫main，叫WinMain，这个函数不同于main，我们不能乱来，它的定义必须与声明保持一致。

## **二、设计与注册窗口类**

设计窗口类，其实就是设计我们程序的主窗口，如有没有标题栏，背景什么颜色，有没有边框，可不可以调整大小等。要设计窗口类，我们用到一个结构——通常情况下，我们用WNDCLASS就可以了，当然还有一个WNDCLASSEX的扩展结构，在API里面，凡是看到EX结尾的都是扩展的意思，比如CreateWindowEx就是CreateWindow的扩展函数。

## **三、创建和显示窗口**

窗口类注册完成后，就应该创建窗口，然后显示窗口，调用CreateWindow创建窗口，如果成功，会返回一个窗口的句柄，我们对这个窗口的操作都要用到这个句柄。什么是句柄呢？其实它就是一串数字，只是一个标识而已，内存中会存在各种资源，如图标、文本等，为了可以有效标识这些资源，每一个资源都有其唯一的标识符，这样，通过查找标识符，就可以知道某个资源存在于内存中哪一块地址中。

## **四、更新窗口（可选）**

为什么更新窗口这一步可有可无呢？因为只要程序在运行着，只要不是最小化，只要窗口是可见的，那么，我们的应用程序会不断接收到WM\_PAINT通知。这里先不说，后面你会明白的。好了，更新窗口，当然是调用UpdateWindow函数。

## **五、消息循环**

Windows操作系统是基于消息控制机制的，用户与系统之间的交互，程序与系统之间的交互，都是通过发送和接收消息来完成的。就好像军队一样，命令一旦传达，就要执行，当然，我们的应用程序和军队不一样，我们收到指令不一要执行，我们是可以选择性地执行。

我们知道，代码是不断往前执行的，像我们刚才写的WinMain函数一样，如果你现在运行程序，你会发现什么都没有，是不是程序不能运行呢，不是，其实程序是运行了，只是它马上结束了，只要程序执行跳出了WinMain的右大括号，程序就会结束了。那么，要如何让程序不结束了，可能大家注意到我们在C程序中可以用一个getchar()函数来等到用户输入，这样程序就人停在那里，直到用户输入内容。但我们的窗口应用不能这样做，因为用户有可能进行其他操作，如最小化窗口，移动窗口，改变窗口大小，或者点击窗口上的按钮等。因此，我们不能简地弄一个getchar在那里，这样就无法响应用户的其他操作了。

可以让程序留在某处不结束的另一个方法就是使用循环，而且是死循环，这样程序才会永久地停在某个地方，但这个死循环必须具有跳出的条件，不然你的程序会永久执行，直达停电或者把电脑砸了。

这样消息循环就出现了，只要有与用户交互，系统人不断地向应用程序发送消息通知，因为这些消息是不定时不断发送的，必须有一个绶冲区来存放，就好像你去银行办理手续要排队一样，我们从最前端取出一条一条消息处理，后面新发送的消息会一直在排队，直到把所有消息处理完，这就是消息队列。

## **六、消息响应**

其实现在我们的应用程序是可以运行了，因为在WindowProc中我们调用了DefWindowProc，函数，消息我们不作任何处理，又把控制权路由回到操作系统来默认处理，所以，整个过程中，我们现在的消息循环是成立的，只不过我们不做任何响应罢了。