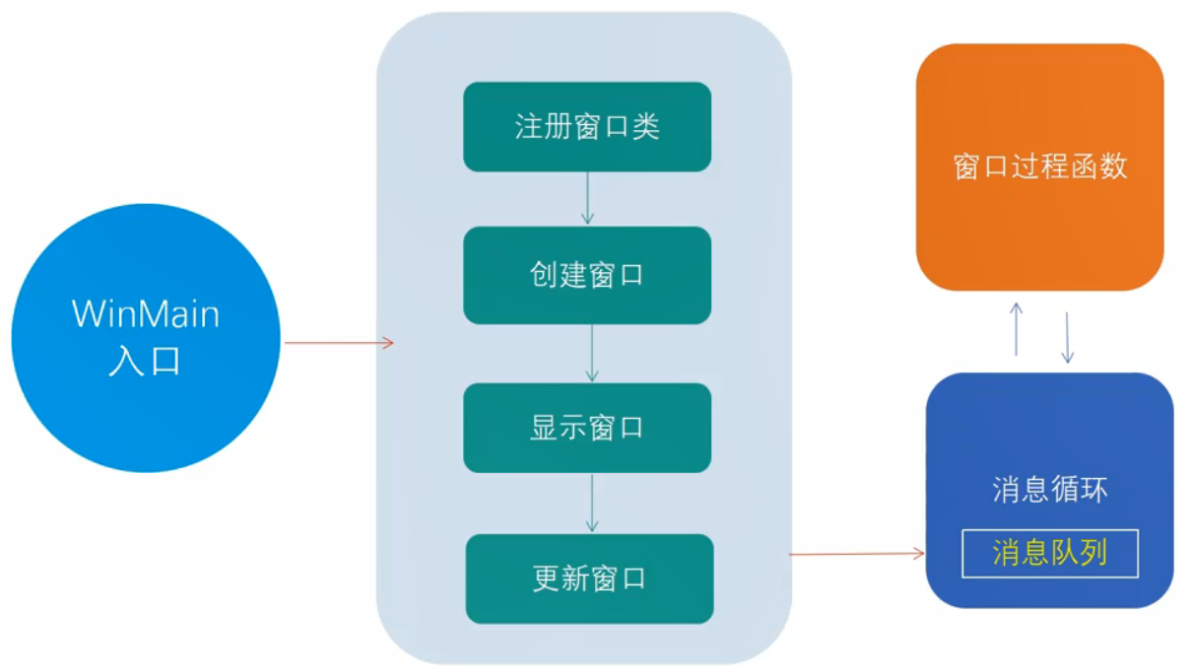


Windows窗口程序

Windows 编程模型

一个完整的Win32程序(`#include <windows.h>`), 该程序实现的功能是创建一个窗口, 并在该窗口中响应键盘及鼠标消息, 程序的实现步骤为:

- WinMain函数的定义
- 创建一个窗口
- 进行消息循环
- 编写窗口过程函数



项目创建

创建新项目

最近使用的项目模板(R)

- Windows 桌面向导 C++
- Windows 桌面应用程序 C++
- 控制台应用 C++
- 空项目 C++
- MFC 应用 C++

搜索模板(Alt+S)(S)

所有语言(L)

所有平台(P)

所有项目类型(T)



空项目

使用 C++ for Windows 从头开始操作。不提供基础文件。

C++ Windows 控制台



控制台应用

在 Windows 终端运行代码。默认打印 "Hello World"。

C++ Windows 控制台



CMake 项目

生成不依赖于 .sln 或 .vcxproj 文件的新式跨平台 C++ 应用。

C++ Windows Linux 控制台



Windows 桌面向导

使用向导自行创建 Windows 应用。

C++ Windows 桌面 控制台 库



Windows 桌面应用程序

具有在 Windows 上运行的图形用户界面的应用程序的项目。

C++ Windows 桌面

Display Name

下一步(N)

配置新项目

Windows 桌面向导

C++ Windows 桌面 控制台 库

项目名称(I)

Demo

位置(L)

D:\MyCode\C++\Demo2021

解决方案(S)

创建新解决方案

解决方案名称(M) ⓘ

Demo

☐ 将解决方案和项目放在同一目录中(D)

Windows 桌面项目

应用程序类型(T)

桌面应用程序(.exe)

其他选项:

☒ 空项目(E)

☐ 预编译标头(P)

☐ 导出符号(X)

☐ MFC 标头(M)

确定

取消

上一步(B)

创建(C)

窗口程序模板代码

```
1 #include <windows.h>
2
3 // 自定义窗口过程回调函数
4 LRESULT CALLBACK MyWindowProc(HWND hwnd, UINT Msg, WPARAM wParam, LPARAM lParam) {
5
6     switch (Msg) {
7     case WM_DESTROY:
8         PostQuitMessage(0);
```

公众号：黑猫编程

网址：<https://noi.hioqier.co>

```

9         return 0;
10    }
11
12    return DefWindowProcA(hwnd, Msg, wParam, lParam);
13 }
14
15 int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR
lpCmdLine, int nShowCmd) {
16
17     // 注册窗口类
18     WNDCLASS wnd;
19
20     wnd.cbClsExtra = 0;
21     wnd.cbWndExtra = 0;
22
23     wnd.hbrBackground = (HBRUSH)(GetStockObject(WHITE_BRUSH));
24     wnd.hCursor = LoadIcon(NULL, IDC_ARROW);
25     wnd.hIcon = LoadIcon(NULL, IDI_APPLICATION);
26     wnd.lpfnWndProc = MyWindowProc;
27     wnd.lpszClassName = TEXT("blackcat");
28     wnd.lpszMenuName = NULL;
29     wnd.style = CS_HREDRAW;
30     wnd.hInstance = hInstance;
31
32     RegisterClassA(&wnd);
33
34     // 创建窗口 返回之前发送 WM_CREATE
35     HWND hwnd = CreateWindowA(
36         TEXT("blackcat"),
37         TEXT("黑猫编程"),
38         WS_OVERLAPPEDWINDOW,
39         100, 100, 300, 300, NULL, NULL, hInstance, NULL
40     );
41
42     // 显示窗口
43     ShowWindow(hwnd, nShowCmd);
44
45     // 更新窗口
46     UpdateWindow(hwnd);
47
48     // 消息循环 收到 WM_QUIT 退出
49     MSG msg;
50     while (GetMessageA(&msg, hwnd, 0, 0)) {
51         TranslateMessage(&msg);
52         DispatchMessageA(&msg);
53     }
54
55     return 0;
56 }

```

