

2、组件对象模型—组件实现1

1. IUnkonwn 接口

a. QueryInterface函数

- i. 带有一个参数，返回值类型是接口类型。

b. AddRef函数和Release函数

- i. 计数器+1 -1

2. 代码介绍

a. Client 调用 impDll 接口

b. 调用 #include <windows.h>

c. ImpDll设置

- i. GameInterface 设置虚函数 `virtual void run() = 0;`
- ii. 设置主函数

C++

```
1  BOOL APIENTRY DllMain( HANDLE hModule,  
2                          DWORD  ul_reason_for_call,  
3                          LPVOID lpReserved  
4                          )  
5  {  
6      return TRUE;  
7  }
```

iii. 设置函数接口

C++

```
1  extern "C" __declspec(dllexport) GameInterface* getGame( )  
2  {  
3      GameInterface* pI = new DiceGame;  
4      return pI;  
5  }
```

1. extern "C" 表示后面的函数用C语言编译，因为C语言和C++的编译方式不一样，C++会把函数名改变
2. __declspec(dllexport) 表示DLL动态链接
3. GameInterface* getGame() 定义指针函数 getGame() 返回 继承的DiceGame的指针

iv. 其他例子

C++

```
1  int num = 0;
2  extern "C" __declspec(dllexport) void setNum( )
3  {
4      cout << "Input an integer number: " ;
5      cin >> num;
6  }
7  extern "C" __declspec(dllexport) int getNum( )
8  {
9      return num;
10 }
```

d. Client设置

i. 定义 接口的约束函数

C++

```
1  class GameInterface
2  {
3  public:
4      virtual void run( ) = 0;
5  };
```

ii. #include <windows.h>

iii. 调用动态链接 ImpDll

1. HINSTANCE Hp = ::LoadLibrary("ImpDll.dll");

iv. 设置本地链接的 接口 GameInterface* (*pFunc)();

v. 接口对接

pFunc = (GameInterface* (*)())::GetProcAddress(Hp,"getGame");

vi. 接口函数调用

C++

```
1 GameInterface* pGame = pFunc( );
2     pGame->run( );
```

vii. 其他例子

C++

```
1 void (*pF1)( );
2 pF1 = (void (*)( ))::GetProcAddress(Hp, "setNum");
3 if (pF1 == NULL)
4 {
5     cout << "Error at calling" << endl;
6     return 3;
7 }
8
9 int (*pF2)( );
10 pF2 = (int (*)( ))::GetProcAddress(Hp, "getNum");
11 if (pF2 == NULL)
12 {
13     cout << "Error at calling" << endl;
14     return 4;
15 }
```

viii. 释放链接 ::FreeLibrary(Hp);

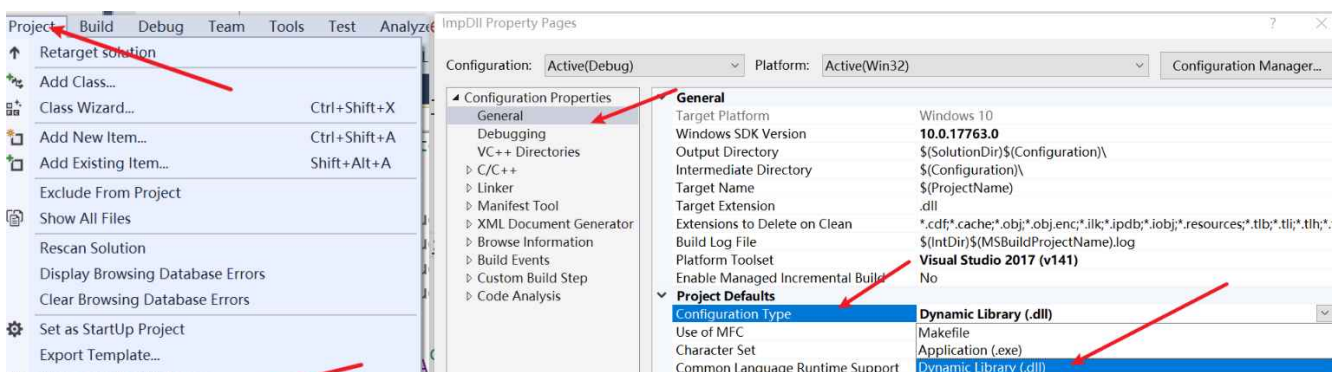
e. 文件设置

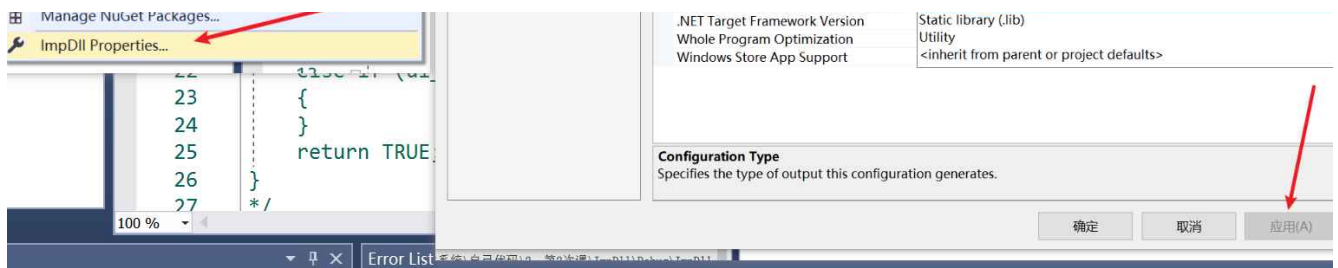
- i. 把 ImpDLL的 bin debug 下的 ImpDll.dll 复制
- ii. 放到 Client 下 bin debug 下 和 exe 同级

3. Visual Studio 2017 运行

a. 生成DLL

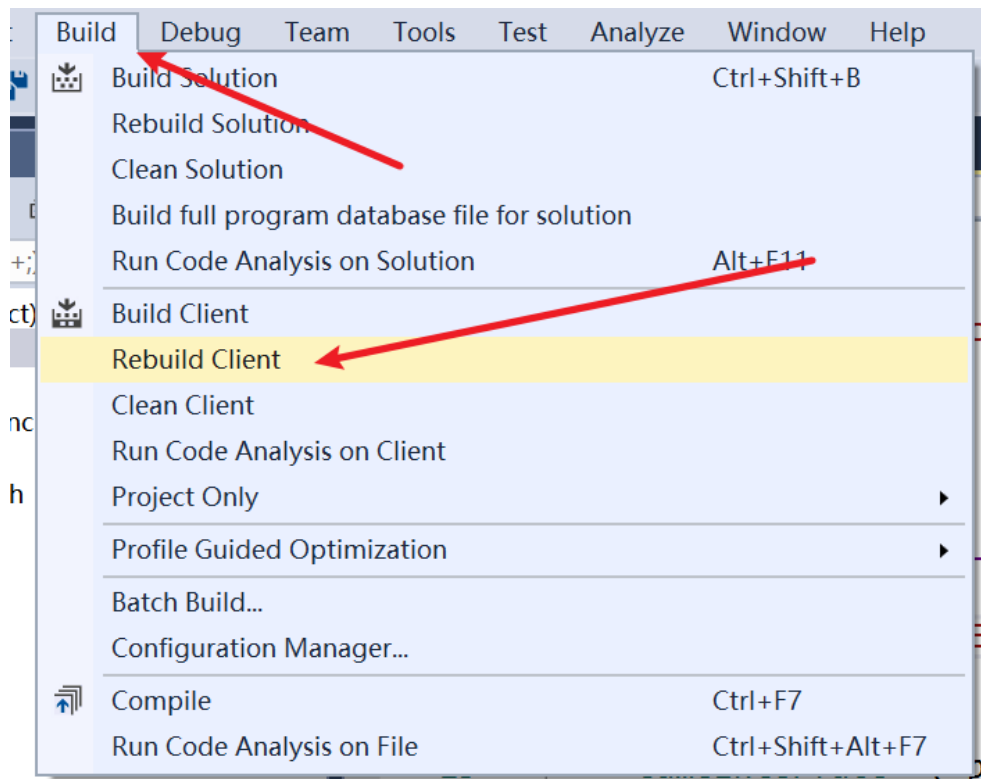
- i. 修改 project projects configuration-projects projects-defaults configuration-type 修改为 dll 点击应用





b. 生成链接 DLL 的客户端

i. Built rebuilt 即可



4. 运行测试

- 把 ImpDLL 的 debug 下的 ImpDll.dll 复制
- 放到 Client 下 debug 下 和 exe 同级
- 双击 Client 下 debug 下的 exe