C#链接数据库问题解决

1.

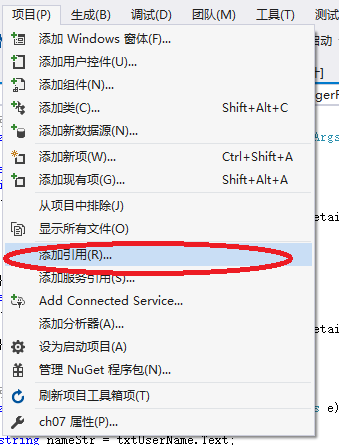
数据库连接方方法可以参考

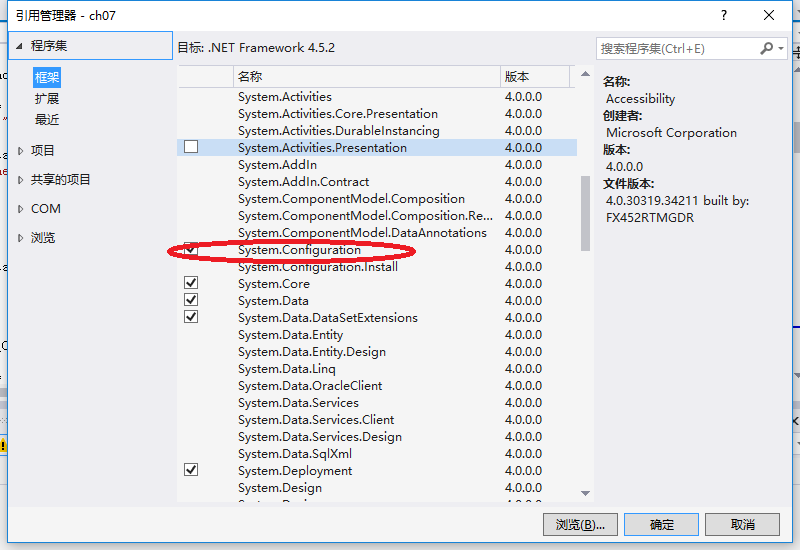
<http://jingyan.baidu.com/article/3ea51489e562bb52e61bbab0.html>

但是在链接数据，写项目时一定要注意

添加system.configuration

添加方法如下





而且还需要用using 导入需要的命名空间

using System.Configuration;

using System.Data.SqlClient;

# [C# Winform 窗体传值 利用委托 子窗体传值给父窗体](http://www.cnblogs.com/xcong/p/3386085.html)

常用的Winform窗体传值有两种方式。

**1.更改Form.designer.cs文件，将控件的设为Public，供子窗体访问。**

　　在designer.cs文件的最后，找到你的控件声明。

private System.Windows.Forms.TextBox textBox1;

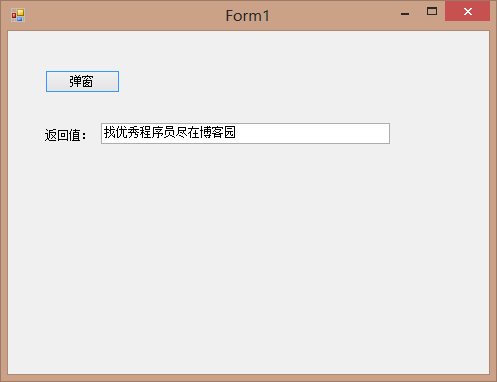
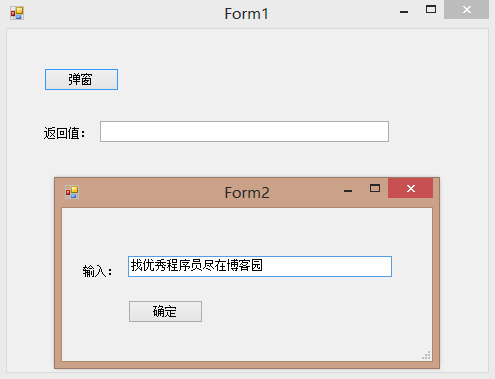
　　更改Private为public，保存即可。

**2.利用委托进行窗体传值。**

父窗体：Form1

　　子窗体：Form2

　　点击Form1，弹出Form2，点击按钮返回值给Form1

****

**首先在Form2中定义委托和事件：**

[复制代码](javascript:void(0);)

//声明委托 和 事件

public delegate void TransfDelegate(String value);

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

public event TransfDelegate TransfEvent;

private void button1\_Click(object sender, EventArgs e)

{

//触发事件

TransfEvent(textBox1.Text);

this.Close();

}

}

[复制代码](javascript:void(0);)

**然后在Form1中进行调用：**

[复制代码](javascript:void(0);)

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Form2 frm = new Form2();

//注册事件

frm.TransfEvent += frm\_TransfEvent;

frm.ShowDialog();

}

//事件处理方法

void frm\_TransfEvent(string value)

{

textBox1.Text = value;

}

}

1. inline是给编译器的建议，函数放在头文件中并在定义时添加，在类声明时直接写的方法体，等同与添加了inline内联；
2. struct、class、enum、union最后面记得加分号，namespace最后没有分号；
3. 默认参数是声明时使用的，默认参数必须在后边；
4. virtual 在父类中声明才有意义，只在子类中声明是错误的。子类中的virtual是可有可无的，一般加上用来标示一下；
5. extern 用在声明时，表示在别处定义了，在.h中定义的函数默认是extern的，但是变量有自己添加；
6. C++03类模板可以有默认参数，但是函数模版不能有默认模版参数，C++0X提供了函数默认模版参数的支持；
7. static在文件中修饰变量说明这个变量在其他文件中是不可见的；
8. switch语句的case只能是常量数字（枚举）或者字符；
9. 有符号的移位运算和无符号的移位运算是不同的，并且分为算术移位和逻辑移位；
10. 宏定义时添加**#define MACRO do{//TODO}while(0)** 来防止产生错误；
11. const的类成员函数在初始化列表中初始化，static的成员函数需要在类外定义初始化，const static可以直接在类中初始化也可以和static一样在类外定义时初始化；
12. 只是返回值不同不能作为重载；
13. 二义性问题需要注意；
14. delete NULL是合法的且不会产生错误的；
15. 对指针进行加减法的步长取决于它指向的类型；
16. 采用加法来交换两个数值容易产生溢出，可以采用异或来做；
17. 直接赋值的char指针是常量字符串，不能修改；
18. 传递进函数的指针内容可以修改，对它本身的修改是错误的，需要时需要传递指针的指针；
19. 注意区别编译器提供的扩展；
20. C++比C有更强的类型检查，有些在C中不需要类型转换的，在C++中需要显示的使用类型转换；
21. 函数返回值和参数的传递都会产生副本；
22. 返回局部变量的引用或者指针是错误的，返回值是正确的，因为它会返回一个副本；
23. 成员函数指针是强类型的，进行转换是需要显示转换，静态成员函数可以直接用普通函数指针来存取；
24. 静态成员函数不能使用virtual、const、volatile修饰，静态函数是类的不存在虚函数表所以不能是virtual的（其他两个不知道为什么）；
25. 静态成员函数只能操作类变量，没有this指针；
26. dynmaic\_cast只支持指针和引用的类型转化，且做运行时类型检测，其他转化不会；
27. sizeof的结果与字节对齐和pack有关；
28. sizeof一个空的类结果是1；
29. 模版的export关键字不被支持，分离定义模型不被支持；
30. goto是个好东西；
31. boost库很好用，但是很容易用错；
32. 自己不能确定的运算符优先级就用括号吧；
33. string::c\_str()返回的是const char\*；
34. 保持函数成员变量声明顺序和初始化列表顺序一致；
35. 只有指针的引用，没有指向引用的指针；
36. 因为c++在c的函数库的名称前做了修改做重载的特性，c++使用c的库时需要在函数前加extern‘C’；
37. const函数只能调用const函数，不能调用非const函数；
38. ->,.,::的优先级高于\*，&；
39. 不要因为编译器支持而省略应该包含的头文件；
40. C中struct、enum、union定义的类型在声明变量时需要这些关键字，而在C++中则不需要；
41. 你可以在C++中嵌套enum来给它加限定，但是在C中这似乎是无效的；
42. enum使用时不要把名称加在前面，enum的成员是全局的;
43. 常量成员函数不能调用非常量成员函数；
44. 常量成员对象只能调用常量成员函数，构造和析构函数除外；
45. 移位运算分为逻辑移位（shl/shr）和算术移位（sar/sal）,而c中的移位运算无符号数是逻辑移位，有符号数是算数移位；
46. 函数重载，模版推演只针对于参数，如果返回值是模版需要显示指定类型;
47. 为了保持可移植，路径使用/来分割，而不是用\;
48. 注意路径和文件的大小写。
49. 嵌套模板时注意用空格分割一下尖括号，即使C++11已经优先解释成嵌套模板了;
50. windows平台使用的wchar\_t可能会在linux平台上有兼容性问题;
51. 只有参数是const的引用才能将返回值直接做参数传递，其他的你都需要为他声明一个变量
52. 有符号数进行类型转换时，是按照高位填充补全的，即最高位是1，填充的高位也会是1，会引起数值的变化;
53. 如果子类存在和基类同名的函数，如果不重载/覆盖就会被隐藏，如果要使用就要显示using Base::fun;
54. 存在多态的关系的类，基类析构函数必须显示定义，并且应为virtual。
55. 含有C++的类型的结构（比如std::string）不要用malloc来分配内存;
56. gcc在windows下使用UNICODE编译，添加-D\_UNICODE的宏定义。