窗体通信

两个窗体之间的关系：

1. A窗体Close后，B窗体Run，非共存关系
2. 在A窗体中的某个方法中New出窗体B，共存关系

情况1的通信方式：

class Aform {

//静态

private static CommunicationData data；

public static void getData( ) { };

public static void setData( ) { };

//会修改通信数据的方法

public void fun( ) {

Aform.setData( );

}

//内部类，用于封装Aform所有对外要通信的数据

class CommunicationData {

A对外通信的所有数据；

}

}

Class Bform {

//Bform使用Aform的通信数据的方法；

Aform.getData; //类名调用静态方法

}

Bform若也需要和Aform通信，那就像Aform一样封装数据类，并提供方法；

情况2的通信方式：

B从A拿数据： Aform通信方法和情况1一样，不过不需要声明为静态static。

A从B拿数据： Bform通信方法和情况2一样，需要声明成静态的。

延伸到类（class）

非共存：A类实例运行完毕并销毁，才开始运行B类

共存主仆：A类实例的方法C内New出一个B类实例。 例子：A a = New A( )； a.C ( ) ;

共存非主仆: A类和B类无丝毫联系但是同时存在。例子：Main{A a = New A( ), B b = New B( ) }

这三种情况的类间通信策略如下：

非共存：采用窗体通信情况1，两个类都定义static通信数据包；

共存主仆：主类通信数据包定义成非静态的，仆类仍旧定义成static。

共存非主仆：两个类的通信数据包都定义成非静态的。

有static变量，肯定存在static构造函数。一个类的构造函数只能初始化自己定义的static变量！