参考：<http://blog.csdn.net/jiaoshiyao/article/details/12091565>

例：程序remove\_delegate\_test1

// Copyright 2016.刘珅珅

// author：刘珅珅

// 移除事件的所有委托

using *System*;

using *System*.*Collections*.*Generic*;

using *System*.*Linq*;

using *System*.*Reflection*;

using *System*.*Text*;

using *System*.*Threading*.*Tasks*;

namespace remove\_delegate\_test1

{

class MyClass

{

public event *EventHandler*<int> SomeEvent;

public event *EventHandler* AnotherEvent;

public void SimulateEvent()

{

if (SomeEvent != null)

SomeEvent(this, 5);

if (AnotherEvent != null)

{

*EventArgs* e = new *EventArgs*();

AnotherEvent(this, e);

}

}

public bool IsSomeEventNull()

{

if (SomeEvent == null)

return true;

return false;

}

public bool IsAnotherEventNull()

{

if (AnotherEvent == null)

return true;

return false;

}

}

class RemoveDelegateTest

{

static void Handler1(object sender, int n)

{

*Console*.*WriteLine*("SomeEvent " + n);

}

static void Handler2(object sender, *EventArgs* e)

{

*Console*.*WriteLine*("AnotherEvent");

}

// 移除事件的委托

static void RemoveEvent<T>(T c, string name)

{

*Delegate*[] invokeList = GetObjectEventList(c, name);

if (invokeList != null)

{

foreach (*Delegate* del in invokeList)

{

*Console*.*WriteLine*("delete Delegate name " + name);

typeof(T).*GetEvent*(name).*RemoveEventHandler*(c, del);

}

}

}

// 获取事件委托列

public static *Delegate*[] GetObjectEventList(object obj, string event\_name)

{

*FieldInfo* field\_info = obj.*GetType*().*GetField*(event\_name,

*BindingFlags*.*NonPublic* | *BindingFlags*.*Instance* | *BindingFlags*.*Public* | *BindingFlags*.*Static*);

if (field\_info == null)

return null;

object field\_value = field\_info.*GetValue*(obj);

if (field\_value != null && field\_value is *Delegate*)

{

*Delegate* delegate\_obj = (*Delegate*)field\_value;

return delegate\_obj.*GetInvocationList*();

}

return null;

}

static void Main(string[] args)

{

MyClass obj = new MyClass();

obj.SomeEvent += Handler1;

obj.AnotherEvent += Handler2;

obj.SomeEvent += Handler1;

obj.AnotherEvent += Handler2;

obj.SimulateEvent();

*Console*.*WriteLine*();

// 删除名称为SomeEvent事件上的所有委托

RemoveEvent<MyClass>(obj, "SomeEvent");

if (obj.IsSomeEventNull())

{

*Console*.*WriteLine*("SomeEvent is null");

}

*Console*.*WriteLine*();

// 删除名称为AnotherEvent事件上的所有委托

RemoveEvent<MyClass>(obj, "AnotherEvent");

if (obj.IsAnotherEventNull())

{

*Console*.*WriteLine*("AnoterEvent is null");

}

}

}

}

输出结果为：

SomeEvent 5

SomeEvent 5

AnotherEvent

AnotherEvent

delete Delegate name SomeEvent

delete Delegate name SomeEvent

SomeEvent is null

delete Delegate name AnotherEvent

delete Delegate name AnotherEvent

AnoterEvent is null