**[常见.NET技术面试题](http://www.cnblogs.com/StanleyCheung/archive/2011/04/22/2024803.html)**

# C#

## 什么是委托？委托和事件是什么关系？

答：简单的说：委托是一种方法容器，里面可以装载若干个具有相同签名的方法引用地址，那么调用委托，就相当于同时调用了该容器内的所有方法。

委托可以将方法作为参数传递给方法.

1. 事件是一种特殊的委托

## 什么是反射？

1. 答: 反射提供了封装程序集，模块和类型对象，可以用反射动态地创建类型的实例，将类型绑定到现有对象，或者从现有对象类型里获取类型，然后调用类型的方法或访问字段和属性。

## C#中是否可以从多个类中继承？如何实现多重继承？

1. 答:不能,要想实现多继承,就用接口

## 什么是密封类？

1. 答: 密封类是类的一种，用sealed修饰，不能用作基类。因此，它也不能是抽象类。密封类主要用于防止派生。

## using关键字有几种用途？

1. 答:两种
2. (1)应用命名空间;
3. (2)定义一个范围,在范围结束时处理对象

## #warning和#error分别的用途是什么？

## string和StringBuilder有什么区别？

1. 答: String 对象是不可改变的。StringBuilder 对象是动态对象

string 是常量，定义后不能改变，

stringbuilder定义后 可以改变，且线程不安全

## const和readonly的区别？

1. 答:const声明常量,raadonly是运行是的常量

## new、virtual、abstract、override、sealed的区别？

1. New :(1)new 一个对象 (2)隐藏基类中的方法(3)在泛型时,new一个泛型类
2. Virtual: virtual 关键字用于修饰方法、属性、索引器或事件声明，并使它们可以在派生类中被重写。virtual 修饰符不能与 static、abstract, private 或 override 修饰符一起使用。

Abstract: **abstract** 修饰符可以和类、方法、属性、索引器及事件一起使用。在类声明中使用 **abstract** 修饰符以指示某个类只能是其他类的基类。标记为抽象或包含在抽象类中的成员必须通过从抽象类派生的类来实现。

1. Override:重写:重写基类中的方法,不能重写非虚方法或静态方法。 重写的基方法必须是 virtual、abstract 或 override 的。override 声明不能更改 virtual 方法的可访问性。 override 方法和 virtual 方法必须具有相同的[访问级别修饰符](http://msdn.microsoft.com/zh-cn/library/wxh6fsc7.aspx)。
2. Sealed:密封类不能作为基类, 密封类不能同时为抽象类

## abstract class和interface有什么区别？

1. 答： 区别一，两者表达的概念不一样。抽象类是一类事物的高度聚合，那么对于继承抽象类的子类来说，对于抽象类来说，属于“是”的关系；
2. 而接口是定义行为规范，因此对于实现接口的子类来说，相对于接口来说，是“行为需要按照接口来完成”。
3. 区别二，抽象类在定义类型方法的时候，可以给出方法的实现部分，也可以不给出；而对于接口来说，其中所定义的方法都不能给出实现部分。
4. 区别三，继承类对于两者所涉及方法的实现是不同的。继承类对于抽象类所定义的抽象方法，可以不用重写，也就是说，可以延用抽象类的方法；
5. 而对于接口类所定义的方法或者属性来说，在继承类中必须要给出相应的方法和属性实现。
 区别四，在抽象类中，新增一个方法的话，继承类中可以不用作任何处理；
6. 而对于接口来说，则需要修改继承类，提供新定义的方法。
7. 如下给出两者的简单对比表格。

|  |  |  |
| --- | --- | --- |
|   | 接口 | 抽象类 |
| 多继承 | 支持 | 不支持 |
| 类型限制 | 没有 | 有，只能是引用类型 |
| 方法实现 | 继承类型中必须给出方法实现 | 继承类中可以不给出 |
| 扩展性 | 比较麻烦 | 相对比较灵活 |
| 多层继承 | 比较麻烦，需要借助虚函数 | 比较灵活 |

## 公共变量与属性的区别？

答: (1)属性只能通过初始化类，创建对象后才能使用。 变量则没有这方面要求，

(2) 属性没有固定的地址，属性的赋值只是使指针;变量赋值只是变量的地址储存所指的地址值

## 结构与类的区别？

答：结构和类具有大体的语法，但是结构受到的限制比类要多。结构不能申明有默认的构造函数，为结构的副本是又编译器创建和销毁的，所以不需要默认的构造 函数和析构函数。结构是值类型，所以对结构变量所做的改变不会影响其的原值，而类是应用类型，改变其变量的值会改变其原值。申明结构用Struck关键 字，申明类用class关键字，向方法传递结构是时是通过值传递的，而不是通过引用。与类不同，结构的实例化可以不使用New关键字。类可以实现接口。

# ASP.NET

## ASP.NET Web应用程序和ASP.NET网站的区别是什么？

答:     都是设计网页的。
都可以添加ASP.Net文件夹

web应用程序Default.aspx显示有两个原有文件及Default.aspx.cs和Default.aspx.designer.cs；Web网站Default.aspx显示有一个原有文件Default.aspx.cs。
web应用程序有重新生成和发布两项；Web网站只有一个发布网站。
web应用程序和一般的winform没有什么区别都有引用的是命名空间等；Web网站在引用后出现一个bin文件夹那里存放dll和pdb文件。
web应用程序可以作为类库被引用；Web网站则不可以作为类库被引用。
web应用程序可以添加ASP.Net文件夹中不包括bin、App\_Code；Web网站可以添加ASP.Net文件夹包括bin、App\_Code。
web应用程序还可添加组件和类；Web网站则没有。
源文件虽然都是Default.aspx.cs但是web应用程序有命名空间，多了一项System.Collections空间引用。

Web应用程序在修改了页面底层的代码之后需要重新进行编译

网站不需要重新编译

网站可以添加APS文件夹，Web应用程序不可以

Web应用程序的项目会生成Dll文件
网站在发布的时候，会自动生成Dll文件，但是每次的Dll文件名都不一样

Web应用程序会有解决方案和项目文件

**WebSite编程模型的优点：**

　　动态编译该页面，马上可以看到效果，不用编译整个站点（主要优势）

　　同上，可以使错误的部分和使用的部分不相干扰（可以要求只有编译通过才能签入）

　　可以每个页面生成一个程序集（不会采用这种方式）

　　可以把一个目录当做一个Web应用来处理，直接复制文件就可以发布，不需要项目文件（无所谓，只适合小站点）

　　可以把页面也编译到程序集中（应该用不到，而且WebApplication也可以通过WebDeployment插件来实现）

## 页面事件的执行顺序？

答: 1.对象初始化, 2.加载视图状态数据, 3.LoadPostData处理回传数据, 4.对象加载5.激发RaisePostDataChanged 事件, 6.处理客户端回传事件, 7.对象预呈现, 8.保存视图状态, 9.呈现HTML, 10.释放

## 页面之间传值有哪几种方法？

答：QueryString,session,cookies,application,server.Transfer,

respose.redictor.

## ASP.NET服务器空件和HTML控件的区别？

答: 1、ASP.NET服务器控件提供了统一的编程接口

       2、ASP.NET服务器控件具有浏览器的无关性，也就是无论何种浏览器都能支持

       3、ASP.NET服务器控件可以保存状态到ViewState里，这样页面在从客户端回传到服务器端或者从服务器端下载到客户端的过程中都可以保存

       4、事件处理模型不同，HTML控件的事件处理都是在客户端的页面上，而ASP.NET服务器控件则是在服务器上

综上所述HTML控件的事件是由客户端页面来触发的，而ASP.NET服务器控件则是由页面把WEB表单发回到服务器端，由服务器来处理

## Server.Transfer和Response.Redirect的区别？

答: Server.Transfer方法把执行流程从当前的ASPX文件转到同一服务器上的另一个ASPX页面。调用Server.Transfer时，当前的ASPX页面终止执行，执行流程转入另一个ASPX页面，但新的ASPX页面仍使用前一ASPX页面创建的应答流。

 如果用Server.Transfer方法实现页面之间的导航，浏览器中的URL不会改变，因为重定向完全在服务器端进行，浏览器根本不知

## XHTML,DHTML与HTML的区别？

XHTML（The Extensible HyperText Markup Language(可扩展标识语言)）实际上是HTML的升级版本。这一版本推出的目的是为了让人们更加容易接受XML，因为XML是为了web发展的方向，本质上说，XHTML是一个过渡技术，结合了部分XML的强大功能及大多数HTML的简单特性。

DHTML：确切地说，DHTML只是一种制作网页的概念，实际上没有一个组织或机构推出过所谓的DHTML标准或技术规范之类的。DHTML不是一种技术、标准或规范，DHTML只是一种将目前已有的网页技术、语言标准整和运用，制作出能在下载后仍然能实时变换页面元素效果的网页的设计概念。

# Winform

## 什么是MDI窗体？

答:父窗体

## 窗体间如何传递数据？

答(1)使用接口传值(2)利用委托(3)使用构造函数

## 进程与多线程有何区别？

答: 进程就是一个单独程序打开了单独的一段地址空间进行单独的工作

而线程是一个程序用N段地址空间进行进行同一种操作

微软官方对进程和线程的定义:

进程：用最简洁的话来说，进程就是一个正在执行的程序，一个或多个线程在进程中运行，线程是操作系统分配CPU运算时间的最小单位。每一个进程都提供了运行一个程序所必需的资源，一个进程具有4GB的虚拟地址空间

## 什么是模态窗口？什么是非模态窗口？

答: 对话框一般分为两种类型：模态类型（modal）与非模态类型（modeless）。所谓模态对话框，就是指除非采取有效的关闭手段，用户的鼠标焦点或者输入光标将一直停留在其上的对话框。非模态对话框则不会强制此种特性，用户可以在当前对话框以及其他窗口间进行切换。

# ADO.NET

## ADO.NET对象模型包括哪几部分？

答: 1.Connection 对象

2.Command对象

3.DataReader对象

4.DataAdapter对象

5.DateSet对象

## 建立与数据库的访问需要几个步骤？

答: 一般要通过以下四个步骤：

　　1、创建一个到数据库的链路，即ADO.Connection;

　　2、查询一个数据集合，即执行SQL，产生一个Recordset;

　　3、对数据集合进行需要的操作;

　　4、关闭数据链路。

## DataAdapter和DataReader有何不同？

答: DataReader对象一般用于以下情况：只想读取数据时；从存储器中获得的数据只用于显示，而不做其他处理。

DataSet对象(与DataAdapter对象一起使用）在以下情况：可能在数据源中修改（编辑或添加新的）数据时；想在读取数据后显示数据前对数据进行修改时。例如，你可能想将数据读入一些表中，或在多个表中的数据间创建关系。你也可能想约列，或者利用多个连接创建多个数据表。

## 事务有哪几个属性？

事务具有ACID属性

　　即 Atomic原子性, Consistent一致性, Isolated隔离性, Durable永久性

　　原子性

　　就是事务应作为一个工作单元,事务处理完成，所有的工作要么都在数据库中保存下来，要么完全回滚，全部不保留

　　一致性

　　事务完成或者撤销后，都应该处于一致的状态

　　隔离性

　　多个事务同时进行，它们之间应该互不干扰.应该防止一个事务处理其他事务也要修改的数据时，不合理的存取和不完整的读取数据

　　永久性

　　事务提交以后，所做的工作就被永久的保存下来

## 在.NET中有几种事务处理方式？

## Oracle和SQL Server在开发上有哪些不同？

答:1操作平台(Oracle可以在多平台上运行,而sql在Windows操作系统上操作)

2.可操作性,易用性(Oracale在8i之前都是dos而sql有界面操作)

3数据库管理(Oracle针对表空间管理,sql针对数据库管理)

4.Oracle没有提供中文帮助文档,而sql提供

5.Oracle的价格要远超于sql(sql是Oracle的十分之一不到)

6.Oracle在Windows操作平台上只有2G空间,而sql就没有限制.但是sql不能运行在其他平台上

# XML

## XML和HTML有什么区别？

## XML操作相关的类有哪些？

## Javascript操作XML的方法？

## ADO.NET与XML如何转换？

## 什么是RSS？如何实现RSS？

# Web Service

## 什么是Web Service？

## Web Service有何优势？

## 如何在ASP.NET中创建Web Service？

## 保证Web Service的安全有哪些措施？

# Ajax

## 什么是Ajax？

## Ajax技术的核心是什么？

## Ajax的工作原理是什么？

## 如何创建ASP.NET Ajax应用程序及常用控件？

## 其他第三方Ajax组建有哪些？

# 系统与文件操作

## 获取系统信息有哪些类？

## 获取系统服务信息需要哪个类？

## 文件操作有哪几个类？

## 注册表操作用到哪些类？

# Network

## .NET网络编程的常用类和方法有哪些？

## 什么是Socket？

## Socket的基本通信过程是怎样的？

## 实现异步Socket通信的关键步骤有哪些？

## TcpListener/TcpClient基本通信过程是怎样的？

# Windows Service

## Windows服务应用程序和普通的应用程序有什么不同？

## Windows服务中的Timer控件和普通Timer控件是否相同？

## 如何安装和卸载Windows服务程序？

## 如何调试Windows服务程序？

# WPF/WCF/LINQ

## .NET 2.0、.NET 3.0和.NET 3.5的区别？

## WPF技术是做什么用的？

## SOA是什么？

## WCF和SOA是什么关系？

## WCF的优势是什么？

## LINQ是什么技术？

## LINQ技术有什么好处？

# OO

## 如何理解面向对象的思想？

## 面向对象的设计与开发会有哪些好处？

## 面向对象有哪些特点？

## 实现多态有哪些方式？

## 在什么情况下应用条件外置和反射技术？

## 如何进行面向对象分析？

## 面向对象设计有哪些原则？

# 三层架构项目开发

## 什么是三层架构？

## 常用三层架构设计是什么？

## 为什么要用三层结构？

## 抽象工厂模式有什么作用？

## 抽象工厂模式的思想方式是什么？

## 条件外置及反射技术的实现是怎样的？

# 大型网站的性能优化与安全

## 如何提高代码的效率和性能？

## 如何提高页面的显示速度？

## 从哪些方面考虑降低服务器负载？

## 实现系统缓存有哪些方法？

## 数据库有哪些常见的优化手段？

## 什么是SQL注入式攻击？

## 在应对大型网站负载架构方面有哪些优化策略？

# 系统设计的原则和技巧

## 在做系统设计时需要考虑哪些方面？

## 开发软件或系统时功能越多越好吗？

## 做项目时，需求总是变化的，你怎么看待这个问题？

## 如何确定软件模块的可变与不变？

## 每个软件在设计时是否都需要考虑软件的可扩展性？

## 对需求或功能比较多的软件或系统应该采取什么样的策略进行开发？

## 什么是软件重构，为什么要重构？

## Web架构设计的周期逻辑是什么？