1. ASP.NET中的身份验证有那些？你当前项目采用什么方式验证请解释  
   ASP.NET身份验证模式包括Windows、Forms（窗体）、Passport（护照）和None（无）。  
       Windows身份验证—常结合应用程序自定义身份验证使用使用这种身份验证模式时，ASP.NET依赖于IIS对用户进行验证，并创建一个Windows访问令牌来表示已通过验证的标识。IIS提供以下几种身份验证机制：λ  
       Passport身份验证。使用这种身份验证模式时，ASP.NET使用Microsoft Passport的集中式身份验证服务，该服务为成员站点提供单一登录和核心配置文件服务。由于访问新的受保护资源或站点时不再需要登录。ASP.NET为Microsoft Passport软件开发包（SDK）所提供的功能提供了一个方便的包装（Wrapper）。此SDK必须安装在WEB服务器上。λ  
       Form窗体身份验证—常用。这种验证方式使用客户端重定向功能，将未通过身份验证的用户转发到特定的登录窗体，要求用户输入其凭据信息（通常是用户名和密码）。这些凭据信息被验证后，系统生成一个身份验证票证（ticket）并将其返回客户端。身份验证票证可在用户的会话期间维护用户的身份标识信息，以及用户所属的角色列表（可选）。λ  
       None。使用这种身份验证模式，表示你不希望对用户进行验证，或是采用自定义的身份验证协议。λ  
   2．什么是WEB控件？使用WEB控件有那些优势？  
   Web 服务器控件定义为抽象控件，是使用于web窗体页并运行于服务器的控件，包括传统的窗体控件，例如按钮、文本框和表等复杂控件,还包括提供在网格中显示数据、选择日期等常用窗体功能的控件。  
   可在服务器上使用熟悉的面向对象技术对其进行编程的对象模型。每个服务器控件都公开一些属性，这些属性使您得以在服务器代码中通过编程操作该控件的 HTML 属性。  
   提供一组事件，您可以为其编写事件处理程序，方法与在基于客户端的窗体中大致相同，所不同的是事件处理是在服务器代码中完成的。  
   在客户端脚本中处理事件的能力。  
   自动维护控件状态。在窗体到服务器往返期间，用户在 HTML 服务器控件中输入的值将在页发送回浏览器时自动维护。  
   与验证控件进行交互，便于您验证用户是否在控件输入了适当的信息。  
   数据绑定到一个或多个控件属性。  
   如果 Web 窗体页显示在支持层叠样式表的浏览器中，则支持 HTML 4.0 样式。  
   直接可用的自定义属性。您可以将任何需要的属性添加到 HTML 服务器控件，页框架将读取并呈现它们而不更改其任何功能。这将允许您向控件添加浏览器特定的属性。  
   Web 服务器控件还提供以下功能：  
       功能丰富的对象模型，该模型具有类型安全编程功能。λ  
       自动浏览器检测。控件可以检测浏览器的功能，并为基本型和丰富型 (HTML 4.0) 浏览器创建适当的输出。λ  
       对于某些控件，可以使用模板来自定义控件的外观。λ  
       对于某些控件，可以指定控件的事件是立即发送到服务器，还是先缓存然后在提交窗体时引发。λ  
       可将事件从嵌套控件（例如表中的按钮）传递到容器控件。λ  
   3．请解释ASP。NET中以什么方式进行数据验证？  
   .net中提供了几个数据验证控件，可以在服务器端或者客户端进行验证。  
   4．请谈谈对正则表达式的看法？  
   正则表达式：Regular Express（也称为“regex”或“regexp”），是一种用来描述文本模式的特殊语法。  
   在编写处理字符串的程序或网页时，经常会有查找符合某些复杂规则的字符串的需要。正则表达式就是用于描述这些规则的工具  
   5．ASP.NET中共有几种类型的控件？各有什么区别？  
   Html控件――传统的html标记+runat=”server”，可在服务器端访问,功能没有Web控件强大  
   Web控件――服务器端控件，可以回传数据，事件驱动  
   用户控件――自定义、可重用的控件，和WebForm很相似，以WebForm为容器运行  
   自定义控件――自定义、可重用的组件，从WebControl继承，在服务器端执行  
   6．WEB控件可以激法服务端事件，请谈谈服务端事件是怎么发生并解释其原理？自动传回是什么？为什么要使用自动传回。  
   通过实现IPostBack这个接口来执行事件。自动回传就是AutoPostBack,使用自动回传可以监视客户端变化情况并将这种改变返回到服务器端  
   7．WEB控件及HTML服务端控件能否调用客户端方法？如果能，请解释如何调用？  
   能，追根溯源，服务器端控件在html中表现形式还是html标记，所以可以执行客户端事件  
   例如：控件.Attributes["客户端事件"]=”客户端脚本”;  
   8．ASP。NET与ASP相比有什么优势？  
   1.开发语言不同 ASP仅局限于使用non-type脚本语言来开发，用户给WEB页中添加ASP代码的方法与客户端脚本中添加代码的方法相同，导致代码杂乱。  
   ASP.NET允许用户选择并使用功能完善的strongly-type编程语言，也允许使用潜加巨大的.NET Framework。  
   2.运行机制不同  
   ASP是解释运行的编程框架，所以执行效率加较低。  
   ASP.NET是编译性的编程框架，运行是服务器上的编译好的公共语言运行时库代码，可以利用早期绑定，实施编译来提高效率。  
   3.开发方式  
   ASP把界面设计和程序设计混在一起，维护和重用困难。  
   ASP.NET把界面设计和程序设计以不同的文件分离开，复用性和维护性得到了提高。  
   9．请解释web.config文件中的重要节点  
   connectionStrings:配置连接字符串，在程序中读取（还可直接提供给其他节点使用，如缓存依赖节点）  
   AppSettings：可以提供用户自己配置参数，然后在程序中读取  
   customErrors：配置程序的错误处理  
   authentication和authorization：配置程序的安全验证和授权  
   trace：配置程序的跟踪  
   globalization：配置程序的全球化，比如：语言  
   pages：配置程序的页设置

10．请解释ASP。NET中的Web页面与其隐藏类之间的关系？  
ASP.NET中的页面由两部分组成，页面显示(aspx文件)和代码逻辑(aspx.cs文件)  
通过Page指令联系在一起，如下：  
<%@Page CodeBehind=”后台代码文件” Inherite=”继承的类”%>  
所以关系为：Web页面会继承隐藏类  
11．什么是ViewState，能否禁用？是否所用控件都可以禁用?  
Web页面是没有状态的，ViewState(视图状态)可用来保存Web页面在客户端和服务器端往返过程的状态  
说简单点ViewState只不过是Hidden控件，加上微软写上的一些代码而已  
可以禁用，由三种方法：  
1)禁用单个控件：控件.EnableViewState = false;  
2)禁用单个页面：<%@ Page EnableViewState=“false”%>  
3)禁用整个程序：web.config中，<Pages enableViewState=“false”>  
12．当发现不能读取页面上的输入的数据时很有可能是什么原因造成的？怎么解决  
可能情况：1)可能事件没有关联  
2)根本就没有写读取代码  
3)写错了事件  
4)在Load事件中有清空代码，但没有写 if (!IsPostBack)  
14．请解释一个WEB页面中代码执行次序。  
常用事件执行次序：Init(初始化)->Load(加载)->Event(控件事件)->PreRender(呈现之前)->UnLoad(卸载)  
15．请解释什么是上下文对象，在什么情况下要使用上下文对象  
HttpContext,封装有关个别 HTTP 请求的所有 HTTP 特定的信息  
无论在Page或者cs中，只要获得有关HTTP有关信息，都可以进行调用  
16．请解释转发与跳转的区别？  
转发:HttpServerUtility.Transfer();  
对于当前请求，终止当前页的执行，并使用指向一个新页的指定 URL 路径来开始执行此新页。可以清除 QueryString 和 Form 集合  
跳转:HttpResponse.Redirect();  
将客户端重定向到新的 URL 并指定该新 URL,默认不终止当前页的执行.  
17．请解释ASP.NET中不同页面之间数据传递有那些方式？  
ASP.NET默认提交地址的是本页,要提交到其他页有以下几种方式:  
1) 会话Session  
写:Session[“name”] = value; 读:Session[“name”]  
2) 应用程序对象Application  
写:Application[“name”] = value; 读:Application[“name”]  
3) 转发:  
第一个页面: Server.Transfer(“WebForm1.aspx”, true);  
第二个页面: Request.Form[“TextBox1”]  
4) 跨页提交:  
第一个页面:指明提交按钮要回发的URL  
< asp:TextBox ID=“txtContent“ runat=”server”/>  
< asp:Button ID=“btnOK” PostBackUrl=“B.aspx” runat=“server” Text=“提交” />  
第二个页面:  
((TextBox)PreviousPage.FindControl(“txtContent”)).Text  
5) QueryString  
第一个页面：URL?名称=值  
第二个页面：Request.QueryString[“名称”]  
6) Cookie  
第一个页面：HttpCookie cookie = new HttpCookie(名称,值); Response.Cookies.Add(cookie);  
第二个页面：Request.Cookies["名称"].Value  
18．请解释ASP.NET中button、linkbutton、imagebutton  及hyperlink这四个控件之间的功别  
Button、LinkButton、ImageButton三个控件比较相似，都可以进行提交并支持了事件驱动，只是样式不同。  
Button为按钮样式，LinkButton为超链接样式，ImageButton为图片样式  
HyperLink为运行在服务器端的超级链接，不支持事件驱动

19．请解释一下。NET多层应用程序中层与层之间以那几种方式进行数据传递。并解释你自己的项目中采用那种方式进行。  
传递的类型不一定，视情况而定，一般可以是DataSet、DataTable、List、Hashtable、泛型等，我一般使用泛型集合List<类型>。

20．如果出现ASP.NET中的事件不能触发可能由于什么原因造成？  
可能的原因有：事件丢失、没有注册事件、客户端验证没有通过等  
21．如果需要在GridView控件中的某一列中添加下拉列表框并绑定数据怎么解决？  
使用模板列ItemTemplate来添加DropDownList  
绑定数据有两种方式：  
1、在GridView执行DataBind()后，for循环FindControl每行的下拉列表并绑定数据  
2、在GridView的RowDataBound事件中，FindControl当前行的下拉列表并绑定数据  
22. BS与CS的联系与区别。  
C/S是Client/Server的缩写。服务器通常采用高性能的PC、工作站或小型机，并采用大型数据库系统，如Oracle、    Sybase、Informix或 SQL Server。客户端需要安装专用的客户端软件。  
B/Ｓ是Brower/Server的缩写，客户机上只要安装一个浏览器（Browser），如Netscape Navigator或    Internet Explorer，服务器安装Oracle、Sybase、Informix或 SQL Server等数据库。在这种结构下，用户界面    完全通过WWW浏览器实现，一部分事务逻辑在前端实现，但是主要事务逻辑在服务器端实现。浏览器通过Ｗ    eb Server 同数据库进行数据交互。  
C/S 与 B/S 区别：  
１．硬件环境不同:  
C/S 一般建立在专用的网络上, 小范围里的网络环境, 局域网之间再通过专门服务器提供连接和数据交换服务.  
B/S 建立在广域网之上的, 不必是专门的网络硬件环境,例与电话上网, 租用设备. 信息自己管理. 有比C/S更强的适应范围, 一般只要有操作系统和浏览器就行  
２．对安全要求不同  
C/S 一般面向相对固定的用户群, 对信息安全的控制能力很强. 一般高度机密的信息系统采用C/S 结构适宜. 可以通过B/S发布部分可公开信息.  
B/S 建立在广域网之上, 对安全的控制能力相对弱, 可能面向不可知的用户。  
３．对程序架构不同  
C/S 程序可以更加注重流程, 可以对权限多层次校验, 对系统运行速度可以较少考虑.  
B/S 对安全以及访问速度的多重的考虑, 建立在需要更加优化的基础之上. 比C/S有更高的要求 B/S结构的程序架构是发展的趋势, 从MS的.Net系列的BizTalk 2000 Exchange 2000等, 全面支持网络的构件搭建的系统. SUN 和IBM推的JavaBean 构件技术等,使 B/S更加成熟.  
４．软件重用不同  
C/S 程序可以不可避免的整体性考虑, 构件的重用性不如在B/S要求下的构件的重用性好.  
B/S 对的多重结构,要求构件相对独立的功能. 能够相对较好的重用.就入买来的餐桌可以再利用,而不是做    在墙上的石头桌子  
５．系统维护不同  
C/S 程序由于整体性, 必须整体考察, 处理出现的问题以及系统升级. 升级难. 可能是再做一个全新的系统  
B/S 构件组成,方面构件个别的更换,实现系统的无缝升级. 系统维护开销减到最小.用户从网上自己下载安    装就可以实现升级.  
６．处理问题不同  
C/S 程序可以处理用户面固定, 并且在相同区域, 安全要求高需求, 与操作系统相关. 应该都是相同的系统  
B/S 建立在广域网上, 面向不同的用户群, 分散地域, 这是C/S无法作到的. 与操作系统平台关系最小.  
７．用户接口不同  
C/S 多是建立的Window平台上,表现方法有限,对程序员普遍要求较高  
B/S 建立在浏览器上, 有更加丰富和生动的表现方式与用户交流. 并且大部分难度减低,减低开发成本.  
８．信息流不同  
C/S 程序一般是典型的中央集权的机械式处理, 交互性相对低  
B/S 信息流向可变化, B-B B-C B-G等信息、流向的变化, 更像交易中心。  
23. 面向对象的特征有哪些方面  
1、封装  
封装是把客观事物封装成抽象的类，并且类可以把自己的数据和方法只让可信的类或者对象操作，对不可信的进行信息隐藏  
2、继承  
继承指可以使用现有类，并在无需重新编写的情况下创建新类的过程  
通过继承创建的新类称为“子类”或“派生类”  
被继承的类称为“基类”、“父类”或“超类”  
3、多态  
是允许你将父对象设置成为和一个或更多的他的子对象相等的技术，赋值之后，父对象就可以根据当前赋值给它的子对象的特性以不同的方式运作。简单的说，就是一句话：允许将子类类型的指针赋值给父类类型的指针。  
25. abstract class和interface有什么区别?  
接口(Interface)是用来定义行为规范的，不会有具体实现，而抽象类除定义行为规范外，可以有部分实现，但一个类能实现多个接口，但只能继承一个父类  
26. 用C＃实现以下功能  
a 产生一个int数组，长度为100，并向其中随机插入1-100，并且不能重复。  
b 对上面生成的数组排序，需要支持升序、降序两种顺序  
int[] iArray= new int[100];  
Random random= new Random();  
ArrayList arraylist= new ArrayList(100);  
for(int i = 0; i < 100; i++){  
arraylist.Add(i);  
}  
for(int i = 0; i < 100; i++){  
int iKey= random.Next(0,99-i);  
IArray[i] = Convert.ToInt32(arraylist[iKey]);  
arraylist.RemoveAt(iKey);  
}  
Array.Sort(iArray);//排序  
Array.Reverse(iArray);//反序

27. 请说出强名的含义  
垃圾回收的原理是根据是否空引用,和该数据类型占用内存来判断该收回多少内存.强类型说白就是必须要有个数据类型. 强名称是由程序集的标识加上公钥和数字签名组成的，其中，程序集的标识包括简单文本名称、版本号和区域性信息（如果提供的话）。您可以确保名称的全局唯一性。  
28. 请列出c＃中几种循环的方法，并指出他们的不同  
1) do: 先循环后判断循环条件  
do{  
// 循环体  
}while(循环条件);  
2) while:根据条件判断是否循环  
While(循环条件)  
{  
循环体  
}  
3) for:可以指定循环次数，递增量等，功能强大  
for(初始值; 条件；递增)  
{  
循环体  
}  
4) foreach:遍历集合，效率最好  
foreach(类型 变量 in 集合)  
{  
循环体  
}  
33. 请指出.net中所有类型的基类  
Object  
34. 请指出GAC的含义  
全局程序集缓存 (global assembly cache, GAC) 。它解决了几个程序共享某一个程序集的问题。不必再将那个被共享的程序集拷贝到应用程序目录了，其实这道理很简单，.net应用程序在加载的时候，会首先查看全局应用程序集缓存，如果有就可以直接使用，没有再到应用程序目录进行查找。  
35. SQL SREVER中，向一个表中插入了新数据，如何快捷的得到自增量字段的当前值  
INSERT INTO jobs (job\_desc,min\_lvl,max\_lvl) VALUES (‘Accountant’,12,125)  
SELECT @@IDENTITY AS ‘Identity’  
37．请解释ASP.NET中以什么方式进行数据验证？  
可以使用验证控件进行客户端和服务器端验证  
也可以使用传统方式：用客户端脚本在客户端验证，用服务器语言在服务器端验证  
38. ArrayList,Array,string[]三者的区别？  
ArrayList 用于不定长的数组，可以随意的增删。  
Array的容量是固定的, 在 Array中，您只能一次获取或设置一个元素的值，支持排序、反转等操作  
Stirng[]和Array比较相似，不支持排序、反转等操作  
39．请说明.net中的错误处理机制，并举例  
异常是从 Exception 类继承的对象，异常从发生问题的代码区域引发，然后沿堆栈向上传递，直到应用程序处理它或程序终止  
try  
{ //执行代码  }  
catch  
{  //出错处理 }  
finally  
{  //无论怎样,都要执行 }  
42. 什么叫应用程序域？  
应用程序域提供安全而通用的处理单元，公共语言运行库可使用它来提供应用程序之间的隔离。您可以在具有同等隔离级别（存在于单独的进程中）的单个进程中运行几个应用程序域，而不会造成进程间调用或进程间切换等方面的额外开销。在一个进程内运行多个应用程序的能力显著增强了服务器的可伸缩性。

43. 常用的调用webservice方法有哪些  
1) 使用WSDL.exe命令行工具。  
2) 使用VS.NET中的Add Web Reference菜单选项  
44. 什么是ASP.net中的用户控件  
用户控件，其实就是网页的一部分，包括 ascx与.cs文件前台标签无素与后台逻辑代码两部分。他不能独立存在，必须用在页面中才有意思。当然，他的目的就是提高页面的相同部分内容的重用  
45. 列举一下你所了解的XML技术及其应用  
XML的应用可分为四类：  
1．可根据客户需要与不同的数据源进行交互。  
2．可将大量运算负荷分布在客户端，即客户可根据自己的需求选择和制作不同的应用程序以处理数据，而服务器只须发出同一个XML文件。  
3．可将同一数据以不同面貌展现给不同的用户。  
4．网络代理可对所取得的信息进行编辑、增减以适应个人用户的需要  
具体热门应用有：ajax技术，WebServices技术。  
46. 值类型和引用类型的区别？写出C#的样例代码。  
值类型：存放的是实际的值，在堆栈中，大多数数据类型、枚举、结构体都是值类型，如：int i = 2;  
引用类型：存放的是地址，在堆中，接口、类、字符串、数组都是引用类型，如：Student s = new Student();  
47. ADO.net中常用的对象有哪些？分别描述一下。  
Connection:连接对象.      传递连接字符串，以连接数据库。  
Command:命令对象          传递 T-sql语句或者存储过程，以执行对数据库的查询与操作。  
DataAdpater:适配器对象     管理 Command对像。主要用于查询或更新数据。  
DataReader:数据读取器对象  快速、以向前、不断开连接的情况下读取数据。  
DataSet:数据集对象         从数据库中取出数据，保存到内存，即可断开连接。  
48． 如何理解委托？  
委托实际上是一种函数指针，指向函数的入口地址。可以动态地将与委托签名（返回类型，参数类型，参数个数）相同的方法赋给一个委托实例，以实现动态调用方法。  
49． C#中的接口和类有什么异同。  
接口里只能有 方法、属性、索引器声明,不能有任何实现。  
其方法的实现，必须由类来完成。其作用是广泛接收实现过它的类的实例，以实现子类的灵活替换。  
类是对一类事物的描述。有属性，有方法，有成员。方法可以有抽像方法，虚方法。抽像方法也是只有声明，没有实现，必段由继承它的子类来重写。虚方法是实现了的方法，只是，在子类，根据需要可重写，也可以不重写。  
50．UDP连接和TCP连接的异同。  
TCP协议是面向连接的，每个数据包的传输过程是：先建立链路、数据传输、然后清除链路。数据包不包含目的地址。受端和发端不但顺序一致，而且内容相同。它的可靠性高，  
UDP协议是面向无连接的，每个数据包都有完整的源、目的地址及分组编号，各自在网络中独立传输，传输中不管其顺序，数据到达收端后再进行排序组装，遇有丢失、差错和失序等情况，通过请求重发来解决。它的效率比较高。  
51. 进程和线程的区别？  
进程是系统进行资源分配和调度的单位；线程是CPU调度和分派的单位，一个进程可以有多个线程，这些线程共享这个进程的资源  
52. 什么是Code-Behind技术。  
就是代码隐藏，在ASP.NET中通过ASPX页面指向CS文件的方法实现显示逻辑和处理逻辑的分离，这样有助于web应用程序的创建。比如分工，美工和编程的可以个干各的，不用再像以前asp那样都代码和html代码混在一起，难以维护。  
52 .net中读写XML的类都归属于哪些命名空间？  
System.Xml;  
53. 解释一下UDDI、WSDL的意义及其作用。  
UDDI: 是Universal Description, Discovery, and Integration的缩写。简单说，UDDI用于集中存放和查找WSDL描述文件，起着目录服务器的作用。  
WSDL:是“Web Services Description Language”的缩写.意如其名,WSDL是Web Services的定义语言。当你实现了某种服务的时候(如,股票查询服务),为了让别的程序调用,你必须告诉大家你的服务的接口.例如,服务名称，服务所在的机器名称，监听端口号，传递参数的类型,个数和顺序,返回结果的类型等等.这样别的应用程序才能调用你的服务。WSDL协议就是规定了有关Web Services描述的标准。  
54. 什么是SOAP,有哪些应用  
SOAP是简单对象访问协议.  
SOAP是消息传递的协议。  
他是WebService技术应用的核心技术之一。它规定了Web Services之间是怎样传递信息的。  
SOAP 是一种轻量级协议，用于在分散型、分布式环境中交换结构化信息。 SOAP 利用 XML 技术定义一种可扩展的消息处理框架，它提供了一种可通过多种底层协议进行交换的消息结构。  
55 .如何部署一个ASP.net页面。  
部署—◊1.网站>指定目录。  这里目的在于：生成 aspx 以及把 cs 文件编译为 .dll文件放到bin目录下面和Web.config  
2.通过IIS.建立一个虚拟目录，指向到这个目录。或者直接把这个目录copy到  wwwroot下面。然后，在IIS中的默认站点中，找到这个目录右击，然后点创建应用程序即可  
56. 简述 private、 protected、 public、 internal 修饰符的访问权限  
private:   私有成员, 在类的内部才可以访问。  
protected: 保护成员，该类内部和继承类中可以访问。  
public:    公共成员，完全公开，没有访问限制。  
internal:   在同一命名空间内可以访问

57. override与重载的区别  
override 与重载的区别。重载是方法的名称相同。参数或参数类型不同，进行多次重载以适应不同的需要  
Override 是进行基类中函数的重写。为了适应需要  
58. 什么是受管制的代码？  
在.net环境中运行的代码都称之为受管制的代码(托管代码)，在.net环境之外，在Windows上运行的代码称之为不受管制的代码  
强类型系统是  
59. 什么是强类型系统？  
RTTI：类型识别系统  
60. 什么是装箱和拆箱？  
装箱是从值类型到引用类型的转换  
拆箱是从引用类型到值类型的转换  
61. CTS、CLS和CLR分别作何解释？  
CTS：通用语言系统。CLS：通用语言规范。CLR：公共语言运行库  
62. String s = new String(“xyz”);创建了几个String Object?  
两个对象，一个是“xyx”,一个是指向“xyx”的引用对象s  
63．什么是强类型，什么是弱类型？哪种更好些？为什么?  
强类型是在编译的时候就确定类型的数据，在执行时类型不能更改，而弱类型在执行的时候才会确定类型。没有好不好，二者各有好处，强类型安全，因为它事先已经确定好了，而且效率高。一般用于编译型编程语言，如c++,java,c#,pascal等,弱类型相比而言不安全，在运行的时候容易出现错误，但它灵活，多用于解释型编程语言，如javascript,vb等  
64．阐述面向接口、面向对象、面向方面编程的区别  
面向接口更关注的是概念，它的原则是先定义好行为规范，再根据行为规范创建实现，严格的来说，面向接口应该是面向对象中的一部分吧，因为面向对象也强调的是依赖倒置原则，也就是实现依赖于抽象，而抽象不依赖于具体实现，更具比较的应该是面向接口与面向抽象对象，我的体会是面向接口更加灵活，但实现时候，稍微有些代码冗余，而面向抽象可以结合面向接口，先定义接口，再定义抽象类，在抽象类中处理一些公共逻辑，再实现具体实现类。面向对象是对复杂问题的分解。面向方面的编程是一种新概念，它解决了很多面向对象无法解决的问题，比如面向对象技术只能对业务相关的代码模块化，而无法对和业务无关的代码模块化。而面向方面正是解决这一问题的方案，它的关键思想是”将应用程序中的商业逻辑与对其提供支持的通用服务进行分离”。  
65．什么是反射  
程序集包含模块，而模块又包括类型，类型下有成员，反射就是管理程序集，模块，类型的对象，它能够动态的创建类型的实例，设置现有对象的类型或者获取现有对象的类型，能调用类型的方法和访问类型的字段属性。它是在运行时创建和使用类型实例  
66. DateTime是否可以为null?  
不能，因为其为Struct类型，而结构属于值类型，值类型不能为null,只有引用类型才能被赋值null  
67. 为什么不提倡catch(Exception)  
原因可能有两点：  
1）try..catch在出现异常的时候影响性能  
2）应该捕获更具体得异常，比如IOExeception,OutOfMemoryException等  
68. 对比抽象基类和接口的使用  
抽象类能有具体实现，而接口只定义行为规范，不能有具体实现。一个类只能继承一个父类，但能实现多个接口  
69. a=10,b=15，在不用第三方变题的前提下，把a,b的值互换  
a=a+b; b=a-b; a=a-b

70. 什么叫做SQL注入，如何防止？请举例说明  
利用sql语言漏洞获得合法身份登陆系统。如身份验证的程序设计成：  
SqlCommand com=new SqlCommand(“Select \* from users where username=’”+t\_name.text+”‘ and pwd=’”+t\_pwd.text+”‘”);  
object obj=com.ExcuteScale();  
if(obj!=null)  
{  
//通过验证  
}  
这段代码容易被sql注入。如用户在t\_name中随便输入，在t\_pwd中输入1′ and 1=’1  就可以进入系统了。  
71. .描述一下C#中索引器的实现过程，是否只能根据数字进行索引？  
可以用任意类型。格式如下：  
修饰符 类型 this [类型 变量]  
{  
get{};  
Set{};  
}  
72. .构造器Constructor是否可被override?  
构造器Constructor不能被继承，因此不能重写Overriding，但可以被重载Overloading  
73. 是否可以继承String类?  
String类是final类故不可以继承  
74. DataReader与Dataset有什么区别？  
一个是只能向前的只读游标，一个是内存中的表。  
75. 什么是虚函数？什么是抽像函数？  
虚函数：可由子类继承并重写的函数。抽像函数：规定其非虚子类必须实现的函数，必须被重写  
76. 在.Net中所有可序列化的类都被标记为?  
[serializable]  
Public class 类名{}  
77. <%# %> 和 <% %> 有什么区别？  
<%# %>表示绑定的数据源，<% %>是服务器端代码块  
78. 能用foreach遍历访问的对象需要实现什么接口或声明什么方法的类型  
IEnumerable接口或 GetEnumerator方法  
79. 叙述Global类中事件发生的顺序  
第一个用户第一次访问第一个页面时，会执行以下的事件  
Application\_Start->Application\_BeginRequest->Session\_Start->Application\_EndRequest  
80. out和ref的区别  
都是引用传递参数，区别  
1、使用ref型参数时，传入的参数必须先被初始化。对out而言，必须在方法中对其完成初始化  
2、使用ref和out时，在方法的参数和执行方法时，都要加Ref或Out关键字。以满足匹配。  
3、out适合用在需要retrun多个返回值的地方，而ref则用在需要被调用的方法修改调用者的引用的时候。  
81. 三层是指什么？为什么要有三层？  
三层指的是：表示层、业务逻辑层和数据访问层。  
表示层：用户界面，和用户进行交互  
业务逻辑层：业务逻辑主要是自定义一些业务规则，完成一些功能。  
数据访问层：和数据库打交道，进行数据的查询和操作  
三层架构主要体现出对程序分而治之的思想，数据访问层只负责提供原始数据，并不需要业务逻辑是什么；业务逻辑层调用数据访问层提供的方法，实现一些业务逻辑，本身不需要了解数据访问层的实现；表示层直接调用业务逻辑提供的方法，把数据呈现给用户，同样，它也不需要了解业务逻辑层和数据访问层是怎么实现的。这样的好处就是降低耦合，易于程序的维护和扩展，每一层都可以由不同的团队来开发，便于分工

82．简单工厂模式有什么作用？  
工厂模式，顾名思义作用于对象的实例化（变化点在”对象创建”,封装”对象创建”）  
我们一般提供一个接口，让该接口负责创建一系列”相关或者相互依赖的对象”，利用反射(Reflection)技术动态的创建实例，无需指定他们具体的类。  
改变配置文件即可方便的控制对象之间的切换  
83. 请编程实现一个冒泡排序的算法  
int num[10] = {1,3,4,6,2,5,7,9,8,0};  
for(int i = 0;i<10;i++)  
{  
for(int j=i+1;j<=10;j++)  
{  
if(num[i]>num[j])  
{  
int temp = num[j];  
num[j] = num[i];  
num[i] = temp;  
}  
}  
}  
84. 用户输入两个字符串str1,str2，返回字符串str1在str2中的出现索引数，如：str1（aba）,str2(2dabab2)返回2，不可以用indexof和substring方法;  
string a = “aba”;  
string b = “2dabab2″;  
bool bFind = false;  
int index = 0;  
while ((index + a.Length) != b.Length){  
if (b.Substring(index).StartsWith(a)){  
bFind = true;  // 找到内容  
break;  
}  
index++;}  
if (bFind){  
Console.WriteLine(“索引在{0}”, index);}  
else{  
Console.WriteLine(“没有包含”);}  
85. C#中的三元运算符  
表达式 ? 结果1(表达式为true时执行) ：结果2(表达式为false时执行)  
86. Sealed修饰符的作用    sealed的中文意思是密封，故名思义，就是由它修饰的类或方法将不能被继承或是重写  
87. 委托的关键字    delegate  
88. 你熟悉的开发模式有哪些  
常用的开发模式有抽象工厂模式、简单工厂模式、工厂方法模式、单例模式、外观模式、代理模式、适配器模式  
89. .NET中，为什么不担心内存漏洞  
垃圾回收器是用来管理应用程序的内存分配和释放的。在垃圾回收器出现以前，程序员在使用内存时需要向系统申请内存空间。有些语言，例如Visual Basic，可以自动完成向系统申请内存空间的工作。但是在诸如Visual C++的语言中要求程序员在程序代码中申请内存空间。如果程序员在使用了内存之后忘了释放内存，则会引起内存泄漏。但是有了垃圾回收器，程序员就不必关心内存中对象在离开生存期后是否被释放的问题。垃圾回收器全部自动帮你完成。  
90. repeater控件的用法  
Repeater没有自己的形态，需要开发人员利用模板进行设置。  
Repeater支持的5种模板分别为ItemTemplate、AlternatingItemTemplate、Separator Template、HeaderTemplate、FooterTemplate  
91. Session里有BUG，微软是怎么处理的  
Session有丢失的可能性，微软对于Session有三种存放方式：  
InProc：session在服务器中以活动对象方式存储（aspnet\_wp.exe）  
StateServer: session被序列化并保存在单独的aspnet\_state.exe的内存中。StateServer能够运行在另一台服务器上  
SQLServer: session被序列化并保存在SQL Server中  
92. Datalist、GridView、Repeater 的区别  
Repeater和DataList比较相似，都没有自己的形式，需要模板的支持  
–Repeater支持ItemTemplate、AlternatingItemTemplate、SeparatorTemplate、HeaderTemplate、FooterTemplate 5种模板  
–DataList除了以上的5种模板，还支持SelectedItemItemplate和EditItemTemplate模板以及支持RepeatColumns(每行呈现的列数)和RepeatDirection(呈现方向)  
GridView有自己的形态，以网格的形式输出内容，功能比Repeater和DataList强

93. AJAX是怎么实现的。用与不用有什么区别  
AJAX实质上是几种技术的集合，包括：XHTML、CSS、DOM、XML、Javascript、XmlHttpRequest。  
主要原理是使用XmlHttpRequest进行异步数据提交，实现局部页面无刷新效果，增加用户体验  
94. 关于CSS的布局。怎么把一个层隐藏？  
Div.style.display = “none”;  
95. C#写连接代码调用存储过程  
SqlConnection conn = new SqlConnection(“…”);  
SqlCommand cmd = new SqlCommand();  
cmd.Connection = conn;  
cmd.CommandType = CommandType.StoredProcedure;  
cmd.CommandText = “存储过程名”;  
conn.Open();  
cmd.ExecuteNonQuery();  
97. 简单谈谈从一个项目启动到客户验收我们所要做的工作步骤  
可行性分析、需要分析。概要设计、详细设计、编码、集成测试、部署  
98. struct 与class的区别 并举例  
1.struct 是值类型，class是对象类型  
2.struct 不能被继承，class可以被继承  
3.struct 默认的访问权限是public,而class默认的访问权限是private.  
4.struct不能由程序员申明构造函数，即使是默认（不带参数）的构造函数，同样也不能有析构的处理部分。这是因为Struct的构造函数是由编译器自动生成的。并且Struct的用途是那些描述轻量级的对象，例如Line,Point等，并且效率比较高。  
5.struct的new和class的new是不同的。struct的new就是执行一下构造函数创建一个新实例再对所有的字段进行Copy。而class则是在堆上分配一块内存然后再执行构造函数，struct的内存并不是在new的时候分配的，而是在定义的时候分配  
99. 把已有的字符串反转，写个方法  
public string Reverse(string str){  
char[] array = str.ToCharArray();  
Array.Reverse(array);  
return new string(array);  }  
100. params关键字的作用  
params 关键字可以指定在参数数目可变处采用参数的方法参数。  
在方法声明中的 params 关键字之后不允许任何其他参数，并且在方法声明中只允许一个 params 关键字。