

选择

1. 静态网页文件的后缀是 C
A. asp B. aspx C. htm D. jsp
2. 在.NET 中 CLS(Commom Language Specification)的作用是 D
A. 存储代码 B. 防止病毒
C. 源程序跨平台 D. 对语言进行规范
3. 在 ASP.NET 中源程序代码先被生成中间代码 (IL 或 MSIL), 然后再转变成各个 CPU 需要的代码, 目的是 C 的需要
A. 提高效率 B. 保证安全
C. 源程序跨平台 D. 易识别
4. .NET 与 XML 紧密结合最大好处是 B
A. 代码易于理解 B. 跨平台传送数据 C. 减少存储空间 D. 代码安全
5. App_Code 目录用来放置中间层的 B
A. 专用数据库文件 B. 共享文件 C. 被保护的文件 D. 代码文件
6. App_Data 目录用来放置 A
A. 专用数据库文件 B. 共享文件 C. 被保护的文件 D. 代码文件
7. 文件系统网站适合学习使用, 因为 D
A. 不要安装 IIS B. 网站允许放置在任意目录下 C. 能够进行单独调试 D. A+B
8. 在网站中, 路径通常有 A 种表示方式, 它们分别是 ()
A. 3 绝对路径 根目录相对路径 文档目录相对路径 B. 2 绝对路径 根目录相对路径
C. 3 绝对路径 根目录绝对路径 文档目录相对路径 D. 2 绝对路径 根目录绝对路径
9. 在 HTML 中, 超链接由 B 标记定义
A. <p> B. <a> C. D. <meta>

10. 下面关于绝对路径的说法，正确的是 D

A. 绝对路径是被连接文档的完整 URL，不包括使用的传输协议 B. 使用绝对路径需要考虑源文件的位置 C. 在绝对路径中，如果目标文件被移动，则链接同样可用 D. 创建外部链接时，必须使用绝对路径

11. HTML `<a name=" NAME" ></a>`表示 D

A. 创建一个超链接

B. 创建一个自动发送电子邮件的链接

C. 创建一个位于文档内部的链接点

D. 创建一个指向位于文档内部的链接点

12. 下面不是 HTML 中表格基本组成部分 C

A. 表行 B. 表列 C. 表框 D. 表题

13. 在网页中可以支持的图像格式是 D

A. PSD B. PSP C. BMP D. GIF, PNG, JPEG

14. 下面几个图形控件中，不能执行鼠标单击事件的控件是 B。

A. ImageButton B. Image C. ImageMap

15. 当需要用控件来输入性别(男、女)或婚姻状况(已婚、未婚)时，为了简化输入，应该选用的控件是 D。

A. RadioButton B. CheckBoxList C. CheckBox D. RadioButtonList

16. Session 与 Cookie 之间最大区别在于 A

A. 存储的位置不同 B. 类型不同

C. 生命周期不同 D. 容量不同

17. 默认情况下 Session 的有效时间是 C

A. 30 秒 B. 10 分钟 C. 20 分 D. 30 分

18. 现在需要验证某个 TextBox 的输入数据是否大于 0，此时应使用的验证控件是 C

A. CompareValidator B. RangeValidator

C. CompareValidator 与 RequiredField Validator

D. RangeValidator 与 RequiredField Validator

19. 现在需要验证某个 TextBox 控件输入的年龄是否大于 18 且小于 65。此时应该使用的验证控件是 D

A. CompareValidator C. RangeValidator B. CompareValidator 与 RequiredField Validator

D. RangeValidator 与 RequiredField Validator

20. ValidatorSummary 验证控件作用是 B

A. 检查总和数 B. 集中显示各个验证结果 C. 判断有无超出范围 D. 检查数值大小

21. 系统提供数据库通用接口目的是 B

A. 提高程序运行的效率 B. 应用程序设计不必考虑数据库的类型

C. 保证程序安全 D. 易于维护

22. XMLDataSource 与 SiteMapDataSource 数据源控件能够用来访问 B

A. 关系型数据 B. 层次型数据

C. 字符串型数据 D. 数值型数据

22. 当进行父、子表之间的同步时，父表与子表记录之间常常是一种 B 的关系

A 多对多 B 一对多 C 一对一 D 多对一

24. 下面是在文件系统网站中父表网页发出的同步指令。代表 D

<http://localhost:3018/Website1/Default2.aspx?Catefory=2>

A 返回的数据 B 打开的新网页

C 传来的参数 D. B+C

25. 在配置 GridView 的 SqlDataSource 数据源控件过程中单击【高级】按钮的目的是 C

- A 打开其他窗口 B 输入新参数
- C 生成 SQL 编辑语句 D 优化代码

26. 在配置 GridView 的 SqlDataSource 数据源控件过程中, 单击【高级】按钮后新打开的窗口中选项显示无效, 是因为 D

A 不能输入参数 B 不能返回数据 C 不能优化代码 D 数据表中缺少关键字段

27. GridView 列模板的作用是 B

- A 增加功能 B A+C
- C 改善数据表的显示 D 定义列格式

28. FormView 与 GridView 最重要的区别 C

- A 能够存储数据 B 外观比较美观
- C 显示布局不受限制 D 数据量受限制

29. 留言板必须加强管理, 这是因为 (D)

- A 存储容量有限 B 查看必须经过授权
- C 影响查看的速度 D 可能存在过时或不健康的留言

30. Wizard 控件的最大特点是 (B)

- A 能够包含大量数 B 能将多个窗口集中到一个网页中
- C 布局不受限制 D 能够快速进行浏览

31. 在 SqlDataSource 数据源控件中, 若将数据库缓存的 CacheExpirationPolicy 属性设置为 Absolute 时, 缓存时限一到则 D

A. 自动延长缓存时间 B. 刷新缓存区并继续缓存 C. 延长一倍缓存时间 D. 缓存区失效

32. 在 SqlDataSource 数据源控件中, 若将数据库缓存的 CacheExpirationPolicy 属性设置为 Sliding 时, 缓存时限一到则 B 。

A. 自动延长缓存时间 B. 刷新缓存区并继续缓存 C. 延长一倍缓存时间 D. 缓存区失效

33. 当一种控件有多种定义时, 用 D 属性来区别它们的定义。

A. ID B. Color C. BackColor D. SkinID

34. 用户控件是后缀为 B 的文件。

A. master B. asax C. aspx D. ascx

35. 主控页是后缀为 A 的文件。

A. master B. asax C. aspx D. ascx

36. 下面是 ASPX 网页中的一段代码:

`<%@PageLanguage="C#"MasterPageFile="~/MasterPage.master">`其中带下划线的部分代表 D 。

A. 主控页的路径 B. 用户控件的名字 C. 用户控件的路径 D. 主控页的名字

37. . 为了保护网页, 需要先将被保护的网页分类放在不同的子目录下, 这是为了 C 。

A. 便于管理 B. 调用方便

C. 集中网页的入口 D. 便于网站迁移

38. . 在一个子目录的 web.config 文件中有如下一段代码:

`<authorization>\\<allow roles="admin" />\\<allow roles="manager" />\\<deny users="*" />\\<allow roles="sales" />\\</authorization>\\`允许访问此子目录下的网页的角色有 C 。

A. admin B. manager C. admin 和 manager D. admin、manager 和 sales

39. 用户登录控件(Login)中的 DestinationPageUrl 属性代表 B 。

A. 登录成功提示 B. 登录成功时转向网页

C. 登录失败转向网页 D. 登录失败提示

40. 内存数据表在创建虚拟购货车中作用 B

A. 网页间共享数据 B. 汇集 临时保存记录 C. 临时保存数据 D. 保存和传送数据

41. Session 在创建虚拟购货车中作用 **D**
- A. 保存数据 B. 汇集记录 C. 临时保存数据 D. 网页间保存和传送数据
42. Page.IsPostBack 属性在 **B** 下为 true。
- A. 新打开网页时 B. 返回本窗体时 C. 关闭网页时 D. 退出网页时
43. 在三层架构中，客户端是系统的前台，负责用户界面的显示；后台负责数据的存储和管理；而中间层负责 **A** 。
- A. 非 UI 的逻辑处理 B. 安全监督 C. 代码优化 D. 协助后台管理
44. 数据表适配器是数据集与数据源之间的桥梁，它的任务是 **D** 。
- A. 将检索后的数据填入数据集 B. 将数据集中更新后的数据返回数据源
- C. 传送用户输入的数据 D. A+B
45. 在中间层中的数据集相当于 **A** 。
- A. 内存中临时的数据库 B. 数据源中的数据表在内存中的副本
- C. 用户输入的数据 D. 受保护的数据
46. 下面是 TreeView 控件的一段代码：
- ```
<asp:TreeNode Text="节点 1" SelectAction="Expand"><asp:TreeNode Text="节点 2"···/><asp:TreeNode Text="节点 3"···/></asp:TreeNode>
```
- 表明用鼠标单击“节点 1”时只进行 **C**
- A. 调用网页    B. 显示或隐藏本节点    C. 折叠或展开子节点    D. 恢复原状

## 填空

1. 动态网页的发展包括以下 CGI 阶段，脚本语言阶段和组件技术几个阶段。
2. .NET 框架由程序设计语言，应用程序平台，ADO.NET 及类库，公共语言运行库和 Visual Studio.NET 开发环境 5 部分组成。
3. .NET 框架中包括一个庞大的类库。为了便于调用，将其中的类按照命名空间进行逻辑分区。

4. ASP.NET 2.0 系统的两层逻辑结构适合于**比较简单的**系统；三层结构适合于**复杂或有特殊要求**的系统。
5. 默认安装中，IIS 服务器被安装在 “[硬盘名]:\ \Inetpub\wwwroot” 的目录下。对应的 URL 是 [http:// Localhost](http://localhost) 或 <http://服务器域名>。
6. 服务器有一个总的配置文件，名为 **Machine.config**。在这个文件中已经确定了所有 ASP.NET 应用程序基本配置。
7. 打开文件系统网站时将自动打开一个**开发服务器**，这是一个轻量级服务器，可以用来对程序进行检查和调试工作。
8. HTML 是 **Hyper Text Markup Language（超文本标记语言）** 的英文缩写。它是 WWW(英文为 **World Wide Web**) 中使用的超文本标记语言。它是由 **Tim Berner-Lee** 在 CERN 发明的。它最早源于 SGML(Standard Generalized Markup Language, 标准通用标记语言)。
9. 对于 HTML 语言，任何**文本编辑器**都可以编辑它。它目前已经成为各种类型**浏览器**的通用标准，它能独立于各种**操作系统**平台。
10. 所有网页，都是由**浏览器**对 HTML 解释而形成的，**浏览器**就相当于 HTML 语言的翻译程序，负责解释 HTML 文件各种符号的含义。
11. 列表文本有**有序列表****无序列表**两种形式
12. 传统的布局方法是**利用表格的定位，对齐等功能**来对网页进行布局；网页布局的最佳方法是**可巧妙地将表格，层和模版结合**。
13. PNG, GI, JPEG 等类型的图像最适合于网页上的实用。这些图像的共同特点是，具有一定的清晰度且**压缩比大，容量小，网上传输和下载的时间短**。
14. IMG 标记有许多属性，其中最常用的是 **SRC, ALT, HEIGHT, WIDTH, BORDERHSPACE, VSPACE, ALIGN**。
15. 图层的作用和最大优点是**图层是一个可以任意移动的容器，甚至允许图层之间重叠放置**。
16. DHTML 的设计思想是：浏览器从服务器端下载文档后，利用**浏览器本身**的资源，在不增加**服务器端**负担和网上传输流量的前提下，使网页的某些元素“动”起来。

17. DHTML 不是一种单一的技术，二十多项技术的综合。在文档对象模型（DOM）的基础上，包括 **HTML4**，**CSS**，**事件**，**脚本**等技术。

18. 文档对象模型 DOM 是 **Document Object Model** 的缩写，它是 **DHTML** 的基础。

19. JavaScript 是由 NetScape 公司开发的一种解释性语言。JavaScript 既可在**服务器**又可在**浏览器**端解释执行，JavaScript 是一种**基于**面向对象和事件驱动的跨平台的**脚本**语言。

20. CSS 全称为 **Cascading Style Sheet**，一般称之为层叠样式表或级联样式表。由 **WorldWideWebConsorti（万维网联盟）**（简称 W3C）制定。简单地说，CSS 就是一组用来控制网页**元素外观**属性。

21. ASPX 网页基类是 **System.Web.UI.Page**

22. ASPX 网页的代码存储模式有两种。它们是**代码分离模式**和**单一模式**。

23. 所谓分布式类就是在多个文件中使用相同的**命名空间**，相同的**类名**，而且每个类的定义前面都加上 **partial** 修饰符，编译时编译器就会自动将这些文件编译成一个完整的类。

24. 若使用 C#语言，在代码分离模式中逻辑代码的文件后缀是 **asp.cs**。

25. 当需要将 TextBox 控件作为密码输入框时，应该将控件的 **TextMode** 属性设置为 Password。

26. 当一个 Web 控件上发生的事件需要立即得到响应时，应该将它的 **AutoPostBack** 属性设为 true。

27. 下面是一个转移到新网页的指令：

```
Response.Redirect(“新网页的 URL”);
```

28. 请将下列数据（nn）在 TextBox 控件中显示出来。double nn=4512.65;

```
TextBox1.Text=nn.ToString();
```

29. 请将下列字符串转换为浮点类型的数据，以便进行计算。string ss=” 4109.54” ;

```
double dd=double.Parse(ss);
```

30. 状态分为 4 种，是：**视图状态**；**应用程序状态**；**会话状态**和 **Cookie 状态**。

31. 下面是设置和取出 Session 对象的代码：设置 Session 的代码是：



```
Session["greeting"]=" Hello Wang!";
```

取出该 Session 对象的语句如下: string MyVar=  
Session[ "greeting" ].ToString();

32. 下面是使用 Application 对象时防止竞争的代码。

```
Application.Lock(); //锁定 Application 对象 Application
["counter"]=(int)Application["counter"]+1;Application.Unlock; //解除对 Application 对象的锁定
```

33. 在浏览器已经封闭 Cookie 的条件下, 为了识别用户应该在应用程序的根目录下的 Web.config 文件中, 对<sessionState>节点做如下配置: <sessionState cookieless=" useUri" />或: <sessionState cookieless=" AutoDelect" >

34. 改变 Session 的有效时间的语句是: Session.Timeout=60。

35. 废除 Session 的语句是: Session.Abandon()。

36. 设计阶段必须将验证控件 ControlTo Validate 属性指向被验证的控件。

37. 是用 RegularExpression 控件验证输入时, 首先要将本控件的 Validation Express 属性设置成检查的模式。

38. 创建数据集的语句是: DataSet ds = new DataSet (); 或者:  
DataSet ds = new DataSet ( "表名" );

39. 数据提供器包括 4 种核心类, 它们是: Connection 连接类、Command 命令类、DataAdapter 数据适配器类 和 DataReader 数据读取类。

40. GridView 控件的基本类是 CompositeDataBoundControl。

41. 分页后每页默认的记录是 10 条。

42. 给待定参数赋值语句中通用格式是: 组件名  
Parameters.Add(newSqlParameter ( "待定参数名", SqlDbType. 类型)); 组件名. Parameters[ "待定参数名" ].Value=实际参数 ;

43. 同一网页中两表同步时, 作为父表的 GridView 中只需增加一个 选择 按钮, 并为被选择的行设置样式。作为子表的 GridView 只需将 父表的 GridView 控件作为提供查询条件的控件即可。

44. 在不同网页中进行同步是, 作为子表的网页应该利用 Request.QueryString() 方法获取从父表传来的同步参数。

45. 网上招聘模板通常分为上、下两部分。上面部分通常使用 FormView 控件的 **Header 模板**，用来介绍公司情况并提出招聘要求；下面部分通常使用 **Insert 模板**，用来输入和存储数据。

46. 当应聘者提交数据后应该在 **ItemInserted** 事件中向应聘者发出感谢或收到的信息。

47. 通常在应聘的界面上要放入“保存”和“复位”两个按钮，其中“复位”按钮采用 **HTNL 的 Reset 按钮** 控件，而“保存”按钮则必须是 **服务器按钮** 控件。

48. 如果要求自动保存提交数据的时间，应该在 **ItemInserting** 事件中编写保存当前时间的代码。

47. 存储过程是用各种 SQL 命令编写并经过编译后直接存放到**数据库端**的程序。

48. 在 SQL Server 数据库中存储过程使用的是 **T\_SQL** 语言。

49. 一个简单的存储过程的代码包括两部分：**CREATE PROCEDURE 存储过程名 ( // 第一部分) AS // 第二部分**其中第一部分是**参数名及类型**；第二部分是 **SQL 语句**。

50. 下面是一段网页缓存的指令：**<%@ OutputCacheDuration="40"VaryByParam=none%>**其中：**Duration="40"**代表**缓存持续时间为 40 秒** **VaryByParam=none** 代表**与版本相关的属性为 non**。

51. 皮肤文件是以 **“.skin”** 为后缀的文件，用来定义**服务器控件**的样式。

52. 下面是一段皮肤文件中的定义：**<asp:TextBox BackColor = "Orange" ForeColor = "DarkGreen" Runat = "server" />**代码将 **TextBox** 服务器控件的底色定义为**橘黄色**，将控件中的字符定义为**暗绿色**。下面是 ASPX 网页中的一段代码：**<%@ Register TagPrefix="uc1"TagName="WebUserControl1"Src="WebUserControl1.ascx" %>**其中带下划线的字符串代表**用户控件的命名空间（前缀）**。

53. Web TreeView 控件的节点的 **EnableClientScript** 属性为 **true** 时，代表允许在浏览器端处理**节点展开和折叠**事件。

54. 当将 TreeView 的 **ExpandDepth** 属性设为 **3** 时，代表默认时显示**深度为 3 层**。

55. **LoginStatus** 控件用来显示用户的**当前的登录状态**，以便随时退出登录状态。

56. **LoginName** 控件用来自动显示登录的**用户姓名**。

57. 当利用 **CreateUserWizard** 创建新用户时，密码不能随便设置，必须符合 **3** 项条件：**7 个字符以上**；**一个字符以上**；**一个以上非数字亦非字母的特殊符号**。

58. 帮助用户恢复密码可以利用 **PasswordRecovery** 控件进行设计。
59. 帮助用户修改密码可以利用 **ChangePassword** 控件进行设计。
60. ASP.NET 2.0 中，中间层中对象都应放入 **App\_Code** 目录中，以便各网页共享
61. **ObjectDataSource** 作为中间层的数据源控件，在访问数据库时能够提供以下几方面的支持：**给数据进行分页和排序；缓存数据；防止数据访问中的冲突。**
62. 在 ASP.NET 2.0 中，系统对个性化服务提供了三方面的支持。这些支持是：**Membership：识别和管理用户信息。Profile：确定和保存用户关心的数据。WebParts：允许用户自己定制网页界面**
63. 为了识别匿名用户需在 Web.config 文件中做如下设置：  
`<system.web><anonymousIdentification enabled= "true" /></system.web>`
64. 如果在 Web.config 文件中定义了一个 Profile `<profile> <property name="Name" /></profile>` 这个 Profile 的名字是 **Name**，类型是 **string**。
65. 为了将定义的 Profile 提供给匿名用户使用，在定义的后面还要添加如下属性：`<profile><properties><add name="PhoneCode" type="string" allowAnonymous= "true" /></properties></profile>`
66. 下面是本章应用程序中生成购货车的部分代码。请填写带字母标号语句的含义：  

```
void GridView1_RowCommand (object sender, GridView Command
EventArgs) {System.Data.DataTable Cart=new System.Data.DataTable(); //
A if (e.CommandName == "buy")

{if(Session["ShoppingCart"]==null) // B

{Cart.Columns.Add("商品编号", typeof (int));Cart.Columns.Add("商品名称",
typeof(string)); ...Session["ShoppingCart"]=Cart;} Cart=(System.Data. Da
taTable)Session["ShoppingCart"]; _____ // C

if (TextBox2.Text == "")

{Validate(); //D} {int index=Convert.ToInt32(e.CommandArgument);GridView
Row row= GridView1.Rows[index];

TextBoxtt=(System.Web.UI.WebControls.TextBox)row.Cells[1].FindControl
("TextBox1"); // E string dgl = tt.Text;int dg = int.Parse(dgl);if (dg
<= 1) dg = 1;
```

```

 string djText = row.Cells[5].Text;

 doubledj=double.Parse(djText); // F

 System.Data.DataRowrr=Cart.NewRow(); //Grr["商品编号"]=bh;...
 Cart.Rows.Add(rr);Session["ShoppingCart"]=Cart; // H}}

```

- A 创建数据表对象 Cart
- B 如果购货车 ShoppingCart2 还没有建立
- C 从 Session 对象中取出数据表
- D 调用校验方法检查是否输入了客户标志
- E 取出 TextBox1 对象
- F 将取出的单价字段转换成浮点类型
- G 将生成新的行对象
- H 将数据表存入 Session 对象中

(2) 下面是本章应用程序中利用 GridView 控件进行汇总计账的代码。请填写带字母标号语句的含义：

```

void Button2_Click(object sender, EventArgs e){double sum=0.0;for(int
ii=0;
ii<GridView1.Rows.Count;ii++)//A{CheckBoxcc=(CheckBox)GridView1.Rows[
ii].Cells[0].FindControl("CheckBox1");//B

if(cc.Checked)//C{sum=sum+(double.Parse(GridView1.Rows[ii].Cells[7].T
ext)); //D}}TextBox2.Text=sum.ToString(); //E}

```

- A 对 GridView 控件中的数据表的记录进行循环
- B 取出记录中的复选框控件
- C 如果复选框控件被选中
- D 求第 8 个字段的总和
- E 将第 8 个字段的总和在 TextBox 控件中显示出来

## 判断

1. 和 ASP 一样，ASP.NET 也是医用基于面向对象的系统（×）
2. asp.net 中能运行的程序语言只有 5 种（×）
3. 在内存管理中垃圾自动回收是指系统对已经不再使用的变量空间自动进行回收（√）
4. XML 语言中的标记由设计者自行定义，用来描述元素的内容（√）
5. XML 是一种编程语言（×）
6. Web.config 是网站中必须的配置文件（×）
7. 网站中的 Global.asax 文件（如果有的话必须放在应用程序的根目录下（√）
8. 离开了网站 ASPX 网页能够单独运行（×）
9. 离开 IIS 服务器.htm 网页能单独运行（√）
10. HTML 语言是对显示的描述，而 XML 语言不仅能够描述显示还能描述语义（√）
11. HTML 语言与 XML 一样，它们的标记都是严格区分大小写的（×）
12. 对于逻辑代码比较复杂的类来说最好采用代码分离模式。（√）
13. 代码分离模式的网页运行效率要高于单一模式。（×）
14. HTML 控件属于浏览器控件，不接受服务器的控制。（√）
15. HTML 控件与 HTML 元素一一对应，而 Web 控件的抽象程度更高，一个控件设置不同的属性时可以实现不同的功能。（√）
16. Web 控件中的几个按钮都可以起到向服务器提交数据的作用。（√）
17. HTML 控件很难转换为服务器控件。（×）
18. HTTP 是一个不包吃状态放入通信协议。这就意味着当浏览器与服务器之间的会话结束，它们之间的连接也就自动断开了，下一次会话与本次连接无关，两次连接之间不存在任何联系（√）
19. 使用 HTML 空间不能保持视图状态（√）
20. 视图状态可以在各个网页之间共享（×）
21. Session 对象可以在同一对话的不同网页之间共享（√）

22. ASP.NET 主要依靠浏览器端对输入进行验证工作，因为浏览器端验证可以将错误发现在提交之前，以减少信息的传输量。（×）

23. 除 RequiredFieldValidator 控件以外，其他验证控件都将被检查对象为空时认为是合法的输入。（√）

24. CompareValidator 控件既可以用来与某个常量比较，也可以用来与另外某个控件的输入进行比较。（√）

25. ADO.NET 只是 ADO 的简单升级（×）。

26. 数据集（DataSet）中可有多张数据表（√）

27. 数据集（DataSet）能够在断开与数据源连接的情况下工作。（√）

28. DataReader 能够在断开与数据源连接的情况下工作。（×）

29. 数据提供器是数据集与数据源联系的中间环节。（√）

30. SqlDataSource 数据源控件只能用于访问 SQL Server 数据库。（×）

31. AccessDataSource 数据源控件只能用于访问 Microsoft Access 数据库。（√）

32. GridView 只能原样显示数据表记录（×）

33. 为了美化显示，GridView 控件的头模板，体模板和尾模板都必须进行设置。（×）

34. 当两张表需要同步时，两张表中都必须有同步字段。同步字段的名称可以不同，但类型必须相同。（√）

35. 利用下拉列表框提供条件进行查询时，下拉列表框 GridView 控件可以共用一个数据源控件。（×）

36. 在同一个网页进行同步时，两个 GridView 控件可以共用一个数据源控件。（×）

37. GridView 是面向记录集合控件，而 DetailsView 是面向单条记录控件。（√）

38. 在 DetailsView 控件中不能显示符合条件的多条记录。（×）

39. 利用 DetailsView 控件增添记录特别方便，因为不需要另外增加输入框。（√）

40. 所有的数据库都可以使用自己定义的存储过程（×）

41. 在 T-SQL 语言中既包括 SQL 语句还可以包括过程语句（√）

- 42. 数据缓存只\能将数据缓存到内存中 (×)
- 43. 主题可为一批服务器控件定义样式 (√)
- 44. 主题目录必须存放在专用目录 App\_Themes 的下面。而皮肤文件必须放在“主题目录”下面 (√)
- 45. 用户控件是一种自定义的组合控件 (√)
- 46. 用户控件不能在同一应用程序的不同网页间重用 (×)
- 47. 使用主控页是为了多个网页在全局的样式上保持一致 (√)
- 48. Web TreeView 控件只能用来描述关系型数据 (×)
- 49. 用 Web TreeView 控件描述的是网站的物理结构 (×)
- 50. Web TreeView 控件节点的 SelectAction 属性中的 Expand 引发的事件通常在服务器中进行处理 (×)
- 51. 利用 SiteMapPath 控件只能显示浏览的当前位置以及经过的路径 (√)
- 52. 所谓角色 (role) 是若干具有相同访问权限用户的集合 (√)
- 53. 只能给每个用户分配一个角色 (×)
- 54. 登录视图控件 (LoginView) 只能有两种模版, 因而只能载入两种视图 (×)
- 55. 数据表适配器采用不同的检索语句, 可以使数据集中获得不同的记录 (√)
- 56. 类库中的类才允许放入中间层中 (×)
- 57. Profiles 是一组为特定用户定义且可以保存的数据 (√)
- 58. 放入 Zone 中的 WebPart 不受限制 (×)
- 59. 能够编辑的 WebPart 类只能放入 EditorZone 中 (√)

1. 动态和静态网页区别①静态网页是指网页内容静态不动, 通常这些网页是用纯粹的 HTML 编写②“动态网页”的动态是指“交互性”, 网页能根据访问者请求或者访问的时间不同而显示不同的内容

2. session 和 application 区别: application 针对所有用户都生效, session 则相反, 每个用户都有自己的 session, 它的生命周期起始于服务器产生对用户请求页面响应, 终止于用户断开与服务器的连接



3. APS.NET 在客户端的状态管理中提供了哪几种方式①视图状态②应用程序状态③Cookie④会话状态

4. 一个 ASP.NET2.0 应用程序由多个文件组成，包括①虚拟目录②网页文件③网页配置文件（Web.config）④网页全局文件（Global.asax）⑤两个共享目录（App\_Code 和 App\_Data）

5. 说明实例变量和静态变量区别。（1）实例变量必须在类进行实例变化后访问，静态变量只能通过类本身访问（2）类在创建他的每一个实例时，都会对实例变量分配内存，不同实例变量使用不同的内存，之间没有任何联系；静态变量只在该变量初始化时进行一次分配，在应用程序中止前均使用统一内存区域。

6. 说明数据绑定控件 FormView 和 DetailsView 的相同点和不同点。相同点：（1）二者每次只显示一个数据进行中的数据，并且具备分页能力（2）支持数据显示，编辑和添加功能。不同点：（1）后者能够自动创建 HTML，表格结构代码，并且只显示相关的数据字段名称和数据值（2）后者能够自动创建表格以显示数据，前者需要开发人员自定义模版属性，以自定义显示各个字段

7. 方法覆盖和方法重载有什么区别。（1）方法重载是指类中两个以上的方法，取相同的方法名，只是使用的参数类型或参数个数不同。（2）方法覆盖是指通过继承类体现，需要在积累中使用 override 关键字修饰即可实现对基类方法的覆盖。派生类的方法应该与覆盖的方法具有同样数量、类型和顺序的参数列表。

## 1. 动态创建一个表格

(1) 先在窗体页中放置新控件的容器。将 Table 控件拖入窗体页中

(2) 将数据绑定到控件上，通常是向表添加 TableCell 控件。然后将单个 TableCell 控件的 Text 属性绑定到数据上，或者向单元格添加数据绑定控件

添加行的方法是：TableRow tRow = new  
TableRow();\\Table1.Rows.Add(tRow); 添加单元格的方法：  
TableCell tCell = new  
TableCell();\\tRow.Cells.Add(tCell);

将【标准】选项卡中的 Table 控件拖入窗体作为动态表格的容器，并为该控件设置相关属性，然后在按钮的 Click 事件中编写代码：

```
public void Button1_Click (object sender, System.EventArgs e){\\int
rowCnt;\\int rowCtr;\\int cellCtr;\\int cellCnt;\\rowCnt =
int.Parse(TextBox1.Text);\\cellCnt =
int.Parse(TextBox2.Text);\\for(rowCtr=1; rowCtr <= rowCnt; rowCtr++)
{\\ TableRow tRow = new TableRow();\\ Table1.Rows.Add(tRow);\\for
```



```
(cellCtr = 1; cellCtr <= cellCnt; cellCtr++)\\ {\\TableCell tCell = new
TableCell();\\tCell.Text = "行 " + rowCtr + ", 列 " + cellCtr;\\
tRow.Cells.Add(tCell);\\}\\}\\}
```

## 2. 在服务器端执行的代码

```
<script runat="server">\\protected void Page_Load(object sender,
EventArgs e)\\ {\\
int\\ttl=int.Parse(DateTime.Now.Hour.ToString());\\if ((ttl >= 6) &&
(ttl <= 12))\\{\\ Response.Write("上午好!");\\}\\else if ((ttl > 12) &&
(ttl <= 18))\\{\\Response.Write("下午好!");\\}\\else\\Response.Write("
晚上好!");\\}\\</script>
```

在浏览器端执行的代码。

```
<script language=javascript >\\{\\var now= new Date();\\var ttl =
now.getHours();\\if ((ttl >= 6) && (ttl <= 12))\\{\\ document.write("
上午好!");\\}\\else if ((ttl > 12) && (ttl <= 18))\\{\\document.write("
下午好!");\\}\\else\\document.write("晚上好!");\\}\\</script>
```

## 3. 学生选课

(1) 建立一新项目，从工具箱向窗体拖入以下控件：2 个 TextBox、3 个 Lable、一个 Button 和一个 DropDownList。

(2) 单机 DropDownList 控件属性 Items 右边的省略号，弹出 ListItem 集合编辑器窗口。

(3) 在 Text 及 Value 中添加下拉列表选项。

(4) 在【设计】视图中双击按钮图标，打开代码文件，在按钮单击事件中编写代码

```
private void Button1_Click(object sender, System.EventArgs
e)\\{\\Response.Write (TextBox1.Text+" 学号: "+TextBox7.Text+"
");\\Response.Write("你选择的课程是: "+
DropDownList1.SelectedValue);\\}
```

## 4. 温度转换

在 ListItem 集合编辑器中添加两个成员（“转换成摄氏” “转换成华氏”），为成员 Text 和 Selected 属性设初值在按钮 Click 事件中写

```
private void Button1_Click(object sender, System.EventArgs e)\\{\\float
rr;\\if (TextBox1.Text=="")\\Response.Write("请输入初始
值。");\\else\\{\\if
(RadioButtonList1.SelectedIndex==0)\\rr=((float.Parse(TextBox1.Text)-
32)*5/9);\\else\\rr=(float.Parse(TextBox1.Text)*9/5+32);\\TextBox2.Te
xt=rr.ToString();\\}\\}
```

## 5. 多图片切换

- (1) 右击项目名，选择【添加】|【添加现有项】，将多张图片加入到项目中。
- (2) 将 RadioButtonList 及图形控件(Image)和按钮(Button)控件从工具箱拖入窗体，单击 RadioButtonList 控件的属性 Items 右边的省略号，弹出【ListItem 集合编辑器】窗口，
- (3) 在窗口中添加项，为每项的 Text、Value 属性赋值，
- (4) 进入 RadioButtonList 控件 SelectedIndexChanged 事件，编写代码：

```
private void RadioButtonList1_SelectedIndexChanged(object sender,
System.EventArgs e)\\
{\\Image1.ImageUrl=RadioButtonList1.SelectedValue;\\}
```

- (5) 运行程序后，改变 RadioButtonList 控件的选择，然后单击按钮，图形控件中的图片将跟随改变。

## 6. 自定义 CustomValidator 控件验证某个输入框输入的数据能否被 3 整除。

```
private void CustomValidator1_ServerValidate (object source,
\\System.Web.UI.WebControls. ServerValidateEventArgs args)\\{\\int
number =int.Parse(args.Value);\\if((number % 3) == 0)
\\args.IsValid=true;\\else\\args.IsValid=false;\\}
```

7. 配置 GridView 控件的数据源控件时选择【高级】按钮就会自动生成“Insert”、“Update”和“Delete”的语句，并且给出各种参数类型的语句。有关“Insert”的语句如下：

```
InsertCommand="INSERT INTO [gyb] ([xm], [xb], [age], [phone],
[ImagePath])\\ VALUES (@xm, @xb, @age, @phone, @ImagePath)"
```

相应的参数类型语句如下：

```
<InsertParameters><asp:ParameterType="String"
Name="xm"></asp:Parameter> <asp:
ParameterType="String"Name="xb"></asp:Parameter><asp:ParameterType="I
```

```

nt32"Name="age"></asp:Parameter><asp:Parameter
Type="String"Name="phone"></asp:Parameter><asp:ParameterType="String"
Name="ImagePath"></asp:Parameter></InsertParameters>

```

8. 如果需要在数据表中增添新记录，则

- (1) 在窗体中增添几个输入框。
- (2) 在 GridView 控件中增添一按钮，并将该按钮的 CommandName 属性命名为 insert。
- (3) 用鼠标双击 RowCommand 事件，书写代码给待定参数赋值。代码如下：

```

void GridView1_RowCommand(object sender, GridViewCommandEventArgs
e){\{\if (e.CommandName == "insert")\{\
SqlDataSource1.InsertParameters.Clear();\

```

```

SqlDataSource1.InsertParameters.Add("xm", TextBox1.Text);\SqlDataSou
rce1.InsertParameters.Add("xb", TextBox2.Text);\SqlDataSource1.InsertP
arameters.Add("age", TextBox3.Text);\SqlDataSource1.InsertParameters.
Add("phone", TextBox4.Text);\SqlDataSource1.InsertParameters.Add("Ima
gePath", TextBox5.Text);\ SqlDataSource1.Insert();\}\}

```

这里的语句中参数都采用 string 类型。即：

```

SqlDataSource1.InsertParameters.Add(string, string);

```

9. Web.config 文件的基本结构

```

<?xml version="1.0" encoding="utf-8"?>\<configuration>\<system.web>
<elementName1>\<childElementName1\attributeName1=value\attributeNa
me2=value\attributeNameN=value/>\</elementName1>\<elementName2\at
tributeName1=value\attributeName2=value\attributeNameN=value/>\...\\
\<elementNameN\attributeName1=value\attributeName2=value\attribute
NameN=value/>\</System.Web>\</configuration>

```

10. 经过连接、分页、排序和选择设置后代码

```

<!--GridView 控件的属性设置-->

```

```

<asp:GridView ID="GridView1"\ Runat=
"server"DataSourceID="SqlDataSource1"DataKeyNames="ProductID"\
AllowPaging="True"\ AllowSorting="True" PageSize="5">

```

```

<!--GridView 中字段属性设置-->

```

```

<Columns>\<asp:CommandField\ShowSelectButton="True"></asp:CommandFi

```

```

eld>\<asp:BoundField\ReadOnly="True"\HeaderText="ProductID"\InsertVisible="False"\DataField="ProductID" SortExpression="ProductID">\</asp:BoundField>\<asp:BoundField\HeaderText="ProductName"\DataField="ProductName"\SortExpression="ProductName">\</asp:BoundField>\<asp:BoundField\HeaderText="CategoryID"DataField="CategoryID"\SortExpression="CategoryID">\</asp:BoundField>\<asp:BoundField\HeaderText="QuantityPerUnit"\DataField="QuantityPerUnit"\SortExpression="QuantityPerUnit">\</asp:BoundField>\<asp:BoundField\HeaderText="UnitPrice"DataField="UnitPrice"\SortExpression="UnitPrice">\</asp:BoundField>\</Columns>\</asp:GridView>

```

```

<!--数据源控件的设置--><asp:
SqlDataSource\ID="SqlDataSource1"\Runat="server"\SelectCommand="SE
LECT[ProductID], [ProductName], [CategoryID], [QuantityPerUnit], \[UnitP
rice]FROM[Products]"\ConnectionString="<%%$\ConnectionStrings:AppCon
nectionString1%">\</asp:SqlDataSource

```

## 11. 参数赋值的语句

```

<asp:SqlDataSource\ID="SqlDataSource1"\
Runat="server"\DeleteCommand=<!--删除命令
-->"DELETEFROM[Products]WHERE
[ProductID]=@original_ProductIDAND[ProductName]=@original_ProductName
AND[CategoryID]=@original_CategoryIDAND[QuantityPerUnit]=@original_Qu
antityPerUnitAND[UnitPrice]=@original_UnitPrice"<!--增加命令
-->InsertCommand="INSERTINTO
[Products]([ProductName], [CategoryID], [QuantityPerUnit], [UnitPrice])V
ALUES (@ProductName, @CategoryID, @QuantityPerUnit, @UnitPrice)" <!--显示
命令-->SelectCommand
="SELECT[ProductID], [ProductName], [CategoryID], [QuantityPerUnit], [Uni
tPrice] FROM [Products]"<!--更新命令-->UpdateCommand
="UPDATE[Products]SET[ProductName]=@ProductName, [CategoryID]=@Categor
yID, [QuantityPerUnit]=@QuantityPerUnit, [UnitPrice]=@UnitPriceWHERE[Pr
oductID]=@original_ProductIDAND[ProductName]=@original_ProductNameAND
[CategoryID]=@original_CategoryIDAND[QuantityPerUnit]=@original_Quant
ityPerUnitAND[UnitPrice]=@original_UnitPrice"ConnectionString="<%%$Con
nectionStrings:AppConnectionString1%">"ConflictDetection="CompareAllVa
lues">

```

```

<!--删除命令中参数--><DeleteParameters>
<asp:ParameterType="Int32"Name="ProductID"></asp:Parameter><asp:Param
eterType="String"Name="ProductName"></asp:Parameter><asp:ParameterTyp
e="Int32"Name="CategoryID"></asp:Parameter><asp:ParameterType="String
"Name="QuantityPerUnit"></asp:Parameter><asp:ParameterType="Decimal"N
ame="UnitPrice"></asp:Parameter></DeleteParameters><!--更新命令中参数

```

-->

```
<UpdateParameters><asp:ParameterType="String"Name="ProductName"></asp:
Parameter><asp:ParameterType="Int32"Name="CategoryID"></asp:Parameter
><asp:ParameterType="String"Name="QuantityPerUnit"></asp:Parameter><a
sp:ParameterType="Decimal"Name="UnitPrice"></asp:Parameter><asp:Param
eterType="Int32"Name="ProductID"></asp:Parameter></UpdateParameters><!
--增加命令中参数--><InsertParameters><asp:Parameter
Type="String"Name="ProductName"></asp:Parameter><asp:ParameterType="I
nt32"Name="CategoryID"></asp:Parameter><asp:ParameterType="String"Nam
e="QuantityPerUnit"></asp:Parameter><asp:ParameterType="Decimal"Name=
"UnitPrice"></asp:Parameter></InsertParameters></asp:SqlDataSource>
```