**实验三 C#和ASP.NET 4.5**

**1.转换输入的成绩到相应的等级**

**（1）控件设置：**

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Grade.aspx.cs" Inherits="Ex3\_Grade" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:TextBox ID="txtInput" runat="server"></asp:TextBox>

<asp:Button ID="btnSubmit" runat="server" OnClick="btnSubmit\_Click" Text="等级" />

<asp:Label ID="lblDisplay" runat="server"></asp:Label>

</div>

</form>

</body>

</html>

**（2）单击事件：**

protected void btnSubmit\_Click(object sender, EventArgs e)

{

float fGrade = float.Parse(txtInput.Text);

int iGrade = (int)(fGrade / 10);

switch (iGrade)

{

case 10:

case 9:

lblDisplay.Text = "优秀";

break;

case 8:

lblDisplay.Text = "良好";

break;

case 7:

lblDisplay.Text = "中等";

break;

case 6:

lblDisplay.Text = "及格";

break;

default:

lblDisplay.Text = "不及格";

break;

}

}

**2.在Web窗体中输出九九乘法表**

**（1）登录事件**

protected void Page\_Load(object sender, EventArgs e)

{

for (int i = 1; i <= 9; i++)

{

for (int j = 1; j <= i; j++)

{

Response.Write(i.ToString() + "x" + j.ToString() + "=" + (i \* j).ToString());

Response.Write("&nbsp;&nbsp");

}

Response.Write("<br/>");

}

}

**3.输入一组以空格间隔的共10个以内的整数，输出该组整数的降序排列**

**（1）控件设置**

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ArrayDescending.aspx.cs" Inherits="Ex3\_ArrayDescending" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:TextBox ID="txtInput" runat="server"></asp:TextBox>

<asp:Button ID="btnSubmit" runat="server" OnClick="btnSubmit\_Click" Text="降序" />

</div>

</form>

</body>

</html>

**（2）单击事件**

protected void btnSubmit\_Click(object sender, EventArgs e)

{

string sInput = txtInput.Text.Trim() + " ";

int j = 0;

int[] aInput = new int[10];

string temp = "0";

for (int i = 0; i <= sInput.Length - 1; i++) {

if (sInput.Substring(i, 1) != " ")

{

temp += sInput.Substring(i, 1);

}

else

{

aInput[j] = int.Parse(temp);

j++;

temp = "0";

}

}

Array.Sort(aInput);

Array.Reverse(aInput);

foreach (int i in aInput)

{

if (i != 0)

{

Response.Write(i+"&nbsp;&nbsp;");

}

}

}

**4.计算两个数的商**

**（1）控件设置**

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Division.aspx.cs" Inherits="Ex3\_Division" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:TextBox ID="txtDivsor" runat="server"></asp:TextBox>

<br />

<asp:TextBox ID="txtDividend" runat="server"></asp:TextBox>

<asp:Button ID="btnSubmit" runat="server" OnClick="btnSubmit\_Click" Text="提交" />

</div>

</form>

</body>

</html>

**（2）单击事件**

protected void btnSubmit\_Click(object sender, EventArgs e)

{

try

{

float divsor = float.Parse(txtDivsor.Text);

float dividend = float.Parse(txtDividend.Text);

Response.Write("商为：" + divsor / dividend);

}

catch (Exception ee)

{

Response.Write("请输入正确的数字！");

}

}

**5.设计并实现一个用户信息类UserInfo**

public class UserInfo

{

private string \_Name;

private DateTime \_Birthday;

public string Name

{

get { return \_Name;}

set { \_Name = value; }

}

public DateTime Birthday

{

get { return \_Birthday; }

set { \_Birthday = value; }

}

public UserInfo(string name,DateTime birthday)

{

this.\_Name = name;

this.\_Birthday = birthday;

}

public string DecideAge()

{

if (DateTime.Now.Year - \_Birthday.Year < 18)

{

return this.\_Name + "，您还没长大呢？";

}

else

{

return this.\_Name + "，您是成人了！";

}

}

}

**6.在Web窗体中应用UserInfo类**

**（1）控件设置**

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="UserInfoPage.aspx.cs" Inherits="Ex3\_UserInfoPage" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:Label ID="Label1" runat="server" Text="姓名："></asp:Label>

<asp:TextBox ID="txtName" runat="server"></asp:TextBox>

<br />

<br />

<asp:Label ID="Label2" runat="server" Text="年龄："></asp:Label>

<asp:TextBox ID="txtBrithday" runat="server"></asp:TextBox>

<br />

<br />

<asp:Button ID="btnSubmit" runat="server" OnClick="btnSubmit\_Click" style="margin-bottom: 0px" Text="提交" />

</div>

</form>

</body>

</html>

**（2）单击事件**

protected void btnSubmit\_Click(object sender, EventArgs e)

{

string name = txtName.Text;

string brithday = txtBrithday.Text;

UserInfo userInfo = new UserInfo(name ,DateTime.ParseExact(brithday,"yyyyMMdd",null));

Response.Write(userInfo.DecideAge());

}

**二、实验拓展：**

**1.扩充成及转换程序。要求增加对输入成绩的合法性判断。**

protected void btnSubmit\_Click(object sender, EventArgs e)

{

try

{

float fGrade = float.Parse(txtInput.Text);

if (fGrade >= 0 && fGrade <= 100)

{

int iGrade = (int)(fGrade / 10);

switch (iGrade)

{

case 10:

case 9:

lblDisplay.Text = "优秀";

break;

case 8:

lblDisplay.Text = "良好";

break;

case 7:

lblDisplay.Text = "中等";

break;

case 6:

lblDisplay.Text = "及格";

break;

default:

lblDisplay.Text = "不及格";

break;

}

}

else

lblDisplay.Text = "请输入正确的成绩！";

}

catch (Exception ee)

{

lblDisplay.Text = "不合法！";

}

}

**2.将九九乘法表改成如图3-13所时的浏览效果。**

protected void Page\_Load(object sender, EventArgs e)

{

for (int i = 1; i <= 9; i++)

{

for (int j = i; j <= 9; j++)

{

Response.Write(i.ToString() + "x" + j.ToString() + "=" + (i \* j).ToString());

Response.Write("&nbsp;&nbsp");

}

Response.Write("<br/>");

}

}

**3.完善实验内容3的程序，要求能完成包含0和负数的排序**

protected void btnSubmit\_Click(object sender, EventArgs e)

{

string sInput = txtInput.Text.Trim() + " ";

int j = 0;int a=0;

for (int x = 0; x <= sInput.Length - 1; x++)

{

if ( sInput.Substring(x, 1) == " ")

{

a++;

}

}

int[] aInput = new int[a];

string temp = " ";

for (int i = 0; i <= sInput.Length - 1; i++)

{

if (sInput.Substring(i, 1) == "-")

{

while (sInput.Substring(i, 1) != " ")

{

temp += sInput.Substring(i, 1); i++;

}

aInput[j] =int.Parse(temp);

j++;

temp = " ";

}

else

{

if (sInput.Substring(i, 1) != " ")

{

temp += sInput.Substring(i, 1);

}

else

{

aInput[j] = int.Parse(temp);

j++;

temp = " ";

}

}

}

Array.Sort(aInput);

Array.Reverse(aInput);

foreach (int i in aInput)

{

if (i != 0)

{

Response.Write(i + "&nbsp;&nbsp;");

}

else

{

if (i == 0)

{

Response.Write("0" + "&nbsp;&nbsp;");

}

}

}

}

**4.使用ArrayList类实现降序排列一组整数的功能**

**ArrayList类：**

public class ArrayList

{

public ArrayList(string sInput)

{

this.\_sInput = sInput;

}

private string \_sInput;

private DateTime \_Birthday;

public string sInput

{

get { return \_sInput;}

set { \_sInput = value; }

}

public string Jx()

{

int j = 0; int a = 0;

for (int x = 0; x <= \_sInput.Length - 1; x++)

{

if (sInput.Substring(x, 1) == " ")

{

a++;

}

}

int[] aInput = new int[a];

string temp = " ";

for (int i = 0; i <= sInput.Length - 1; i++)

{

if (sInput.Substring(i, 1) == "-")

{

while (sInput.Substring(i, 1) != " ")

{

temp += sInput.Substring(i, 1); i++;

}

aInput[j] = int.Parse(temp);

j++;

temp = " ";

}

else

{

if (sInput.Substring(i, 1) != " ")

{

temp += sInput.Substring(i, 1);

}

else

{

aInput[j] = int.Parse(temp);

j++;

temp = " ";

}

}

}

Array.Sort(aInput);

Array.Reverse(aInput);

string s = "";

foreach (int i in aInput)

{

if (i != 0)

{

s+=i + "&nbsp;&nbsp;";

}

else

{

if (i == 0)

{

s+="0" + "&nbsp;&nbsp;";

}

}

}

return s;

}

}

**设计页面**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class Ex3\_ArrayDescending : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void btnSubmit\_Click(object sender, EventArgs e)

{

string sInput = txtInput.Text.Trim() + " ";

ArrayList arraylist = new ArrayList(sInput);

Response.Write(arraylist.Jx());

}

}

**6.改写UserInfo类，要求如下**

**（1）增加一个事件ValidateBrithday。**

**（2）改写DecideAge（）方法，当输入的生日值大于当前日期或小于1900-1-1时触发事件ValidateBrithday**

**（3）设计页面并应用修改后的UserInfo类。**

**改写的UserInfo类：**

public class UserInfo

{

private string \_Name;

private DateTime \_Birthday;

public string Name

{

get { return \_Name;}

set { \_Name = value; }

}

public DateTime Birthday

{

get { return \_Birthday; }

set { \_Birthday = value; }

}

public UserInfo(string name,DateTime birthday)

{

this.\_Name = name;

this.\_Birthday = birthday;

}

public delegate void EventHandler(object sender, EventArgs e);//声明事件所需的代理

public event EventHandler ValidateBrithday;

public void OnValidateBrithday(object sender, EventArgs e)

{

if (ValidateBrithday != null)

{

ValidateBrithday(this, e);

}

}

public string DecideAge()

{

if (DateTime.Now.Year- \_Birthday.Year<0|| \_Birthday.Year<1900)

{

OnValidateBrithday(this, EventArgs.Empty);

return" ";

}

else

{

if (DateTime.Now.Year - \_Birthday.Year < 18)

{

return this.\_Name + "，您还没长大呢？";

}

else

{

return this.\_Name + "，您是成人了！";

}

}

}

}

**设计页面**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class Ex3\_UserInfoPage : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

private void V(object sender, EventArgs e)

{

Response.Write("请输入合法的生日！");

}

protected void btnSubmit\_Click(object sender, EventArgs e)

{

string name = txtName.Text;

string brithday = txtBrithday.Text;

UserInfo v = new UserInfo(name, DateTime.ParseExact(brithday, "yyyyMMdd", null));

v.ValidateBrithday += new UserInfo.EventHandler(V);

v.DecideAge();

UserInfo userInfo = new UserInfo(name,DateTime.ParseExact(brithday,"yyyyMMdd",null));

Response.Write(userInfo.DecideAge());

}

}