V. Windows Forms – Validation, Exceptions, ListView, TreeView

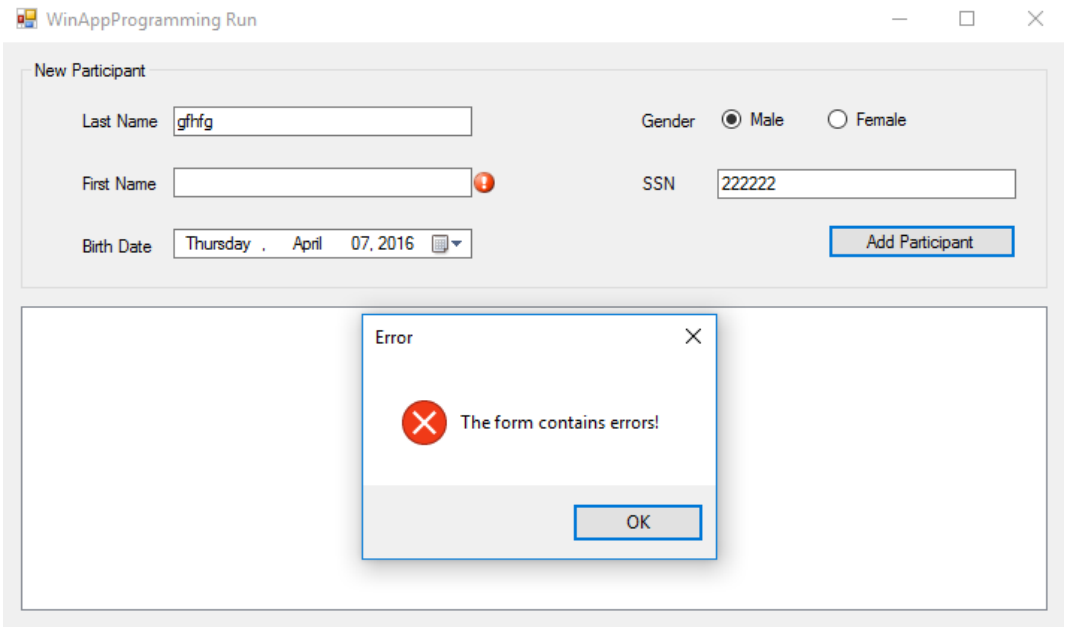
Contents

1. Data Validation

Assignment

C# Sample code available at http://online.ase.ro – "ValidationCustomExceptions" Sample

1. Create a new project with the name "ValidationCustomExceptions"
2. Create the following UI
3. Add ErrorProviders for the LastName and FirstName fields



2. Exception Handling

2.1. Custom Exceptions

Assignment

C# Sample code available at http://online.ase.ro – "ValidationCustomExceptions" Sample

1. Add the following "InvalidBirthDateException" class

```
public class InvalidBirthDateException : Exception
{
	public DateTime BirthDate { get; set; }

	public InvalidBirthDateException(DateTime birthDay)
	{
		BirthDate = birthDay;
	}

	public override string Message
	{
		get
		{
			return "The birthDate " + BirthDate + " is invalid";
		}
	}
}
```

2. Update the "BirthDate" property in the "Participant" class in order to validate the received value

```
#region BirthDate
private DateTime _birthDate;
public DateTime BirthDate {
	get { return _birthDate; }
	set
	{
		if(value >= DateTime.Today)
			throw new InvalidBirthDateException(value);
		_birthDate = value;
	}
}
#endregion
```

3. Update the event handler for the "Add Participant" button in order to handle the potential exceptions.

```
try
{
	var participant = new Participant(lastName, firstName, birthDate, gender, ssn);
	//TODO Logic for adding the participant to the list bellow
}
catch (InvalidBirthDateException ex)
{
	//Expected exception
	MessageBox.Show(string.Format("The birth date {0} is invalid!", ex.BirthDate));
}
catch (Exception)
{
```

```
        //UnExpected exception
        MessageBox.Show("An exception has been encountered! Please contact the technical
support.");

        //Log the exception using:
        // - Log4Net
        // - Application Insights
}
finally
{
        Debug.WriteLine("Always executed");
}
```

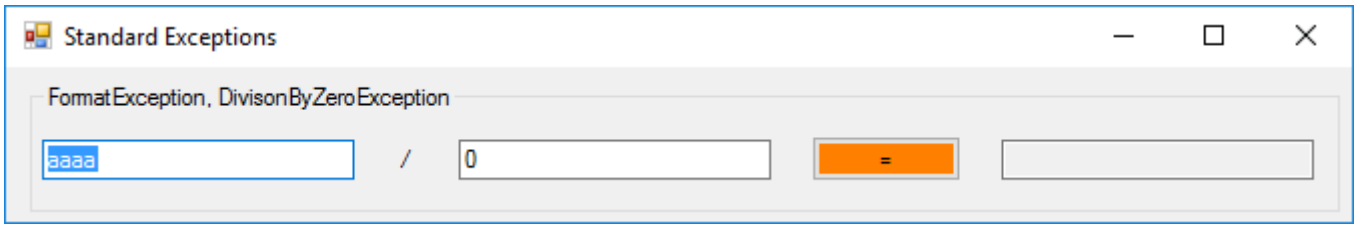
2.2. Standard Exceptions

- common exception types: System.NotImplementedException, System.DivideByZeroException, System.FormatException

Further reading: link

Assignment

1. Create a new project with the name "StandardExceptions"
2. Create the following UI



3. Handle the possible exceptions

```
try
{
        int value1 = int.Parse(tbValue1.Text);
        int value2 = int.Parse(tbValue2.Text);

        tbResult.Text = (value1/value2).ToString(CultureInfo.InvariantCulture);

        //Throwing an exception:
        //throw new NotImplementedException();
}
catch (FormatException ex)
{
        MessageBox.Show(ex.Message);

        //ReThrowing the exception
        throw; //Handled by Program.Application_ThreadException
}
catch (DivideByZeroException ex)
{
        MessageBox.Show(ex.Message);
}
catch (Exception ex)
{
```

```
        MessageBox.Show(ex.Message);
}
```

4. Catch all application exceptions

```
 // Add at the begining of the Main method
Application.ThreadException += Application_ThreadException;

private static void Application_ThreadException(object sender,
System.Threading.ThreadExceptionEventArgs e)
{
        Debug.WriteLine(e.Exception.Message);
}
```

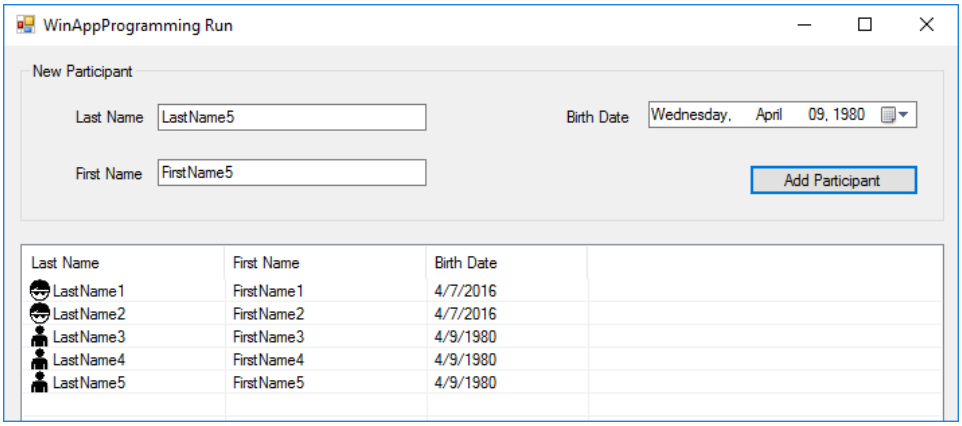
3. Complex Visualization Controls

3.1. ListView

Assignment

C# Sample code available at http://online.ase.ro – "ListViewSample" Sample

1. Create a new project with the name "ListViewSample"
2. Rename "Form1" to "MainForm"
3. Create the following UI



4. Add the Participant class

```
internal class Participant
{
      public string LastName { get; set; }
      public string FirstName { get; set; }
      public DateTime BirthDate { get; set; }


      public Participant(string lastName, string firstName, DateTime birthDate)
      {
            LastName = lastName;
            FirstName = firstName;
```

```
            BirthDate = birthDate;
        }
}
```

4. Final form of the “MainForm” class

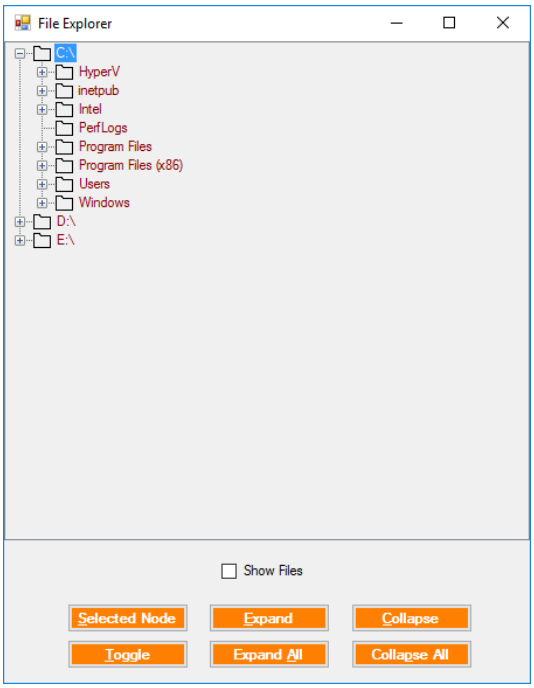
```
public partial class MainForm : Form
{
        #region Properties
        private List<Participant> Participants { get; set; }
        #endregion

        public MainForm()
        {
                InitializeComponent();

                Participants = new List<Participant>();
        }

        public void DisplayParticipants()
        {
                lvParticipants.Items.Clear();

                foreach (Participant participant in Participants)
                {
                        var listViewItem = new ListViewItem(participant.LastName);
                        listViewItem.SubItems.Add(participant.FirstName);
                        listViewItem.SubItems.Add(participant.BirthDate.ToShortDateString());

                        //approximate calculation of the age
                        if ((DateTime.Now - participant.BirthDate).TotalDays / 365 >= 18)
                                listViewItem.ImageKey = "adult.png";
                        else
                                listViewItem.ImageKey = "child.png";

                        lvParticipants.Items.Add(listViewItem);
                }
        }

        #region Events
        private void btnAdd_Click(object sender, EventArgs e)
        {
                string firstName = tbFirstName.Text;
                string lastName = tbLastName.Text;
                DateTime birthDate = dtpBirthDate.Value;

                var participant = new Participant(lastName, firstName, birthDate);
                Participants.Add(participant);

                DisplayParticipants();
        }
        #endregion
```

3.2. TreeView

Assignment

C# Sample code available at http://online.ase.ro – “TreeViewSample” Sample

1. Create a new project with the name “TreeViewSample”
2. Create the following UI



3. Add the following method

```
 #region Methods
private void FillDirectoryTree()
{
      //  Suppress redraw until tree view is complete
      tvw.BeginUpdate();

      //  First clear all the nodes.
      tvw.Nodes.Clear();

      //  Get the logical drives and put them into the root nodes.
      //  Fill an array with all the logical drives on the machine.
      string[] strDrives = Environment.GetLogicalDrives();

      //  Iterate through the drives, adding them to the tree.
      //  Use a try/catch block, so if a drive is not ready,
      //  e.g. an empty floppy or CD, it will not be added to the tree.
      foreach (string rootDirectoryName in strDrives)
      {
            try
            {
                  //  Find all the first level subdirectories.
                  //  If the drive is not ready, this will throw an
                  //  exception, which will have the effect of
                  //  skipping that drive.
                  Directory.GetDirectories(rootDirectoryName);

                  //  Create a node for each root directory
                  TreeNode ndRoot = new TreeNode(rootDirectoryName);

                  //  Add the node to the tree
                  tvw.Nodes.Add(ndRoot);
```

```
                    //  Add subdirectory nodes.
                    //  If Show Files checkbox checked, then also get the filenames.
                    GetSubDirectoryNodes(ndRoot, cb.Checked);
              }
              catch (IOException)
              {
                    //  let it through
              }
              catch (Exception e)
              {
                    //  Catch any other errors.
                    MessageBox.Show(e.Message);
              }
        }

        tvw.EndUpdate();

}

private void GetSubDirectoryNodes(TreeNode parentNode, bool getFileNames)
{
        //  Exit this method if the node is not a directory.
        DirectoryInfo di = new DirectoryInfo(parentNode.FullPath);
        if ((di.Attributes & FileAttributes.Directory) == 0)
        {
              return;
        }

        //  Clear all the nodes in this node.
        parentNode.Nodes.Clear();

        try
        {
              //  Get an array of strings containing all the subdirectories in the
parent node.
              string[] arSubs = Directory.GetDirectories(parentNode.FullPath);

              //  Add a child node for each subdirectory.
              foreach (var subDir in arSubs)
              {
                    DirectoryInfo dirInfo = new DirectoryInfo(subDir);
                    // do not show hidden folders
                    if ((dirInfo.Attributes & FileAttributes.Hidden) != 0)
                    {
                          continue;
                    }

                    TreeNode subNode = new TreeNode(dirInfo.Name);
                    subNode.ImageIndex = 0;
                    subNode.SelectedImageKey = "openFolder.png";
                    parentNode.Nodes.Add(subNode);
              }

              if (getFileNames)
              {
                    //  Get any files for this node.
                    string[] files = Directory.GetFiles(parentNode.FullPath);

                    // After placing the nodes,
                    // now place the files in that subdirectory.
                    foreach (string str in files)
                    {
```

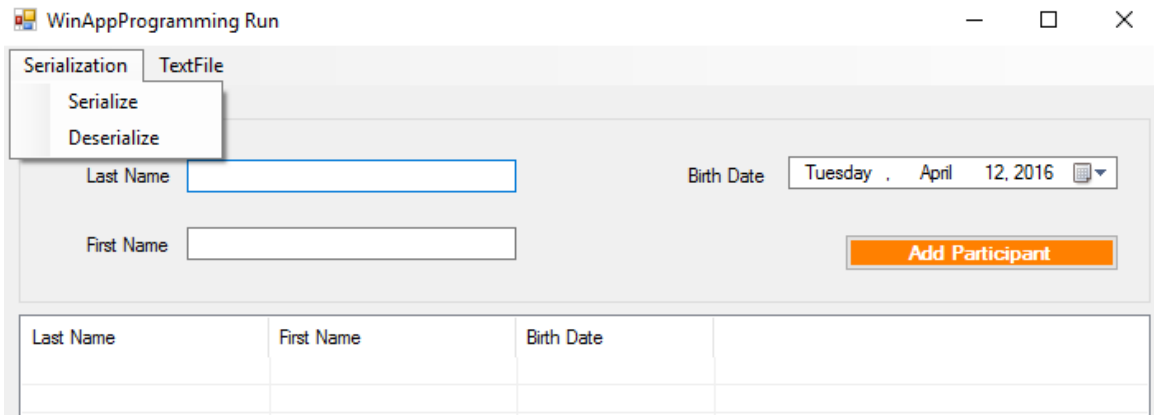
```
                        FileInfo fi = new FileInfo(str);
                        TreeNode fileNode = new TreeNode(fi.Name);
                        parentNode.Nodes.Add(fileNode);

                        //  Set the icon
                        switch (fi.Extension.ToUpper())
                        {
                              case ".JPG":
                              case ".JPEG":
                                    fileNode.ImageKey = "jpgFile.png";
                                    fileNode.SelectedImageKey = "jpgFile.png";
                                    break;
                              case ".TXT":
                                    fileNode.ImageKey = "textFile.png";
                                    fileNode.SelectedImageKey = "textFile.png";
                                    break;
                              default:
                                    fileNode.ImageKey = "file.png";
                                    fileNode.SelectedImageKey = "file.png";
                                    break;
                        }
                  }
            }
      }
      catch (UnauthorizedAccessException)
      {

      }
}
#endregion
```

4. Working with Files

4.1. Serialization/Deserialization

Activity

1. Add the options to Serialize, Deserialize and export as a Text File, as shown bellow



5. Secondary Forms

Assignment

1. Change the default startup form