عنوان: **ساخت فرم های generic در WPF و Windows Application**
نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۲/۰۴/۰۹ ۸:۳۰
آدرس: www.dotnettips.info
برچسب‌ها: WPF, Generics

آیا می‌توان در یک پروژه های Windows App یا WPF، یک فرم پایه به صورت generic تعریف کنیم و سایر فرم‌ها بتوانند از آن ارث ببرند؟ در این پست به تشریح و بررسی این مسئله خواهیم پرداخت.
در پروژه هایی به صورت Smart UI کد نویسی شده اند و یا حتی قصد انجام پروژه با تکنولوژی‌های WPF یا Windows Application را دارید و نیاز دارید که فرم‌های خود را به صورت generic بسازید این مقاله به شما کمک خواهد کرد.

Windows Application#

یک پروژه از نوع Windows Application ایجاد می‌کنیم و یک فرم به نام FrmBase در آن خواهیم داشت. یک Label در فرم قرار دهید و مقدار Text آن را فرم اصلی قرار دهید.
در فرم مربوطه، فرم را به صورت generic تعریف کنید. به صورت زیر:

```
public partial class FrmBase<T> : Form where T : class
    {
        public FrmBase()
        {
            InitializeComponent();
        }
    }
```

بعد باید همین تغییرات را در فایل FrmBase.designer.cs هم اعمال کنیم:

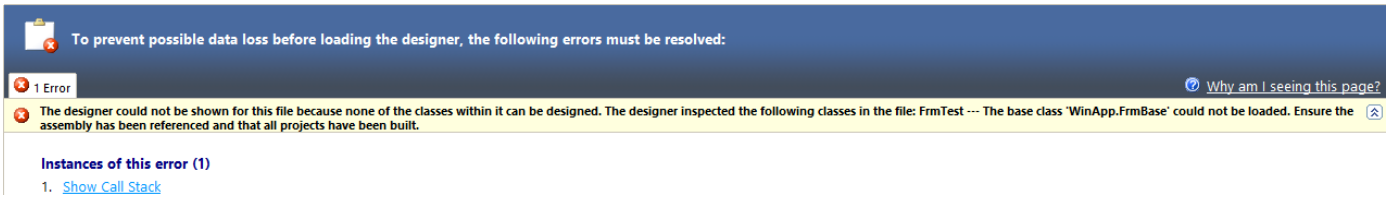
```
partial class FrmBase<T> where T : class
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose( bool disposing )
        {
            if ( disposing && ( components != null ) )
            {
                components.Dispose();
            }
            base.Dispose( disposing );
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.label1 = new System.Windows.Forms.Label();
            this.SuspendLayout();
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Location = new System.Drawing.Point(186, 22);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(51, 13);
            this.label1.TabIndex = 0;
            this.label1.Text = "فرم اصلی";
            //
            // FrmBase
```

```
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(445, 262);
            this.Controls.Add(this.label1);
            this.Name = "FrmBase";
            this.Text = "Form1";
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.Label label1;
    }
```

یک فرم جدید بسازید و نام آن را FrmTest بگذارید. این فرم باید از FrmBase ارث ببرد. خب این کار را به صورت زیر انجام می‌دهیم:

```
public partial class FrmTest : FrmBase<String>
    {
        public FrmTest()
        {
            InitializeComponent();
        }
    }
```

پروژه را اجرا کنید. بدون هیچ گونه مشکلی برنامه اجرا می‌شود وفرم مربوطه را در حالت اجرا مشاهده خواهید کرد. اما اگر قصد باز کردن فرم FrmTest را در حالت design داشته باشید با خطای زیر مواجه خواهید شد:



با این که برنامه به راحتی اجرا می‌شود و خروجی آن قابل مشاهده است ولی امکان نمایش فرم در حالت design وجود ندارد. متاسفانه در Windows Appها برای تعریف فرم‌ها به صورت generic یا این مشکل روبرور هستیم. تنها راه موجود برای حل این مشکل استفاده از یک کلاس کمکی است. به صورت زیر:

```
public partial class FrmTest : FrmTestHelp
    {
        public FrmTest()
        {
            InitializeComponent();
        }
    }

    public class FrmTestHelp : FrmBase<String>
    {
    }
```

مشاهده می‌کنید که بعد از اعمال تغییرات بالا فرم FrmTest به راحتی Load می‌شود و در حالت designer هم می‌تونید از آن استفاده کنید.

WPF#

در پروژه‌های WPF ، راه حلی برای این مشکل در نظر گرفته شده است. در WPF، برای Window یا UserControl پایه نمی‌توان

Designer داشت. ابتدا باید فرم پایه را به صورت زیر ایجاد کنیم:

```
public class WindowBase<T> : Window where T : class
{
}
```

در این مرحله یک Window بسازید که از WindowBase ارث ببرد:

```
public partial class MainWindow: WindowBase<String>
    {
        public MainWindow()
        {
            InitializeComponent();
        }
    }
```

در WPF باید تعاریف موجود برای Xaml و Code Behind یکی باشد. در نتیجه باید تغییرات زیر را در فایل Xaml نیز اعمال کنید:

```
<local:WindowBase x:Class="GenericWindows.MainWindow"

        x:TypeArguments="sys:String"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:local="clr-namespace:GenericWindows"
        xmlns:sys="clr-namespace:System;assembly=mscorlib"
        Title="MainWindow" Height="350" Width="525">
    <Grid>

    </Grid>
</local:WindowBase>
```

همان طور که می‌بینید در ابتدای فایل به جای Window از local:WindowBase استفاده شده است. این نشان دهنده این است که فرم پایه برای این Window از نوع WindowBase است. برای مشخص کردن نوع generic هم می‌تونید از x:TypeArguments استفاده کنید که در این جا نوع آن را String انتخاب کردم.

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۴/۰۹ ۱۹:۶

با تشکر. لطفا یک مثال دنیای واقعی از این فرم جنریک بزنید.

نویسنده: مسعود م.پاکدل
تاریخ: ۱۳۹۲/۰۴/۱۰ ۹:۳۷

یک مثال پیاده سازی شده رو می‌تونید (^) اینجا مشاهده کنید.

نویسنده: محمدی راوری
تاریخ: ۱۳۹۲/۰۵/۰۹ ۱۴:۲۴

با سلام و تشکر از آموزش ارائه شده
من در ساخت برنامه مشکلی نداشتم و اون رو ساختم اما در مرحله ای که لازم بود تا نمایش فرم ساخته شده را فعال کنم با مشکل برخوردم.
اگر ممکنه راهنمایی کنید؛ با سپاس فراوان.

نویسنده: مسعود م.پاکدل
تاریخ: ۱۳۹۲/۰۵/۰۹ ۲۰:۲۶

مشکل در قسمت نمایش در حالت Design بوده است یا اجرا؟
اگر امکانش هست مشکل مربوطه را دقیق عنوان کنید.