عنوان: **به دام انداختن خطاهای مدیریت نشده در برنامه‌های Windows forms دات نت**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۰/۱۴ ۱۴:۰۳:۵۱
آدرس: www.dotnettips.info
برچسب‌ها: Windows forms

شبیه به نحوه‌ی به دام انداختن خطاهای مدیریت نشده در Web forms و روال استاندارد Application_Error ، در برنامه‌های Windows forms نیز این امر به صورت زیر ممکن است:

```
using System;
using System.Threading;
using System.Windows.Forms;

namespace testWinForms87
{
 static class Program
 {
    /// <summary>
    /// The main entry point for the application.
    /// </summary>
    [STAThread]
    static void Main()
    {
        // handling UI thread exceptions
        Application.ThreadException += uIThreadException;

        // force all Windows Forms errors to go through our handler.
        Application.SetUnhandledExceptionMode(UnhandledExceptionMode.CatchException);

        // handling non-UI thread exceptions.
        AppDomain.CurrentDomain.UnhandledException += currentDomainUnhandledException;


        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new Form1());
    }

    private static void currentDomainUnhandledException(object sender, UnhandledExceptionEventArgs e)
    {
        MessageBox.Show(((Exception)e.ExceptionObject).Message, "currentDomainUnhandledException");
    }

    private static void uIThreadException(object sender, ThreadExceptionEventArgs e)
    {
        MessageBox.Show(e.Exception.Message, "uIThreadException");
    }
 }
}
```

چند نکته:
الف)همانطور که ملاحظه می‌کنید سطرهای فوق باید قبل از Application.Run در روال اصلی برنامه تعریف شوند.
ب) این متدها استاتیک هستند و توصیه شده است در پایان برنامه ارجاعات آنها را حذف کنید تا نشتی حافظه رخ ندهد. دقیقا به همین صورت =+ که اضافه شدند با =- هم قابل حذف هستند.
ج) در حالت اجرا شدن uIThreadException ، برنامه بسته نخواهد شد (و بدیهی است در صورت عدم بکار گیری این روش، حتما برنامه کرش خواهد کرد). برای مثال شاید علاقمند نباشید که بخاطر عدم دسترسی نوشتن در پوشه‌ای خاص، خطای حاصل سبب بسته شدن کل برنامه شود. به این صورت این موارد را می‌توان به دام انداخت. اما currentDomainUnhandledException که حاصل از خطاهای ایجاد شده برای مثال در یک ترد دیگر بجز ترد اصلی برنامه هستند، حتما سبب بسته شدن برنامه خواهند شد. بنابراین اینجا تنها شانس لاگ کردن خطای مدیریت نشده حاصل را خواهیم داشت. به همین منظور همیشه توصیه می‌شود که در تردهای ایجاد شده در برنامه، حتما موارد مدیریت خطاها را لحاظ نمائید، زیرا خطاهای حاصل شده در آن‌ها غیرقابل اغماض بوده و حتما سبب کرش برنامه می‌شوند.

پ.ن.

دقیقا در برنامه‌های Win32 دلفی هم چنین قابلیتی به همین شکل و تقریبا با همین نام‌ها وجود دارد. فقط کافی است روالی را جهت Application.OnException ایجاد کنید: ;)

```
procedure TmyFrmMain.FormCreate(Sender: TObject);
begin
  Application.OnException := MyExceptionHandler;
end;
procedure TmyFrmMain.MyExceptionHandler(Sender: TObject; E: Exception);
begin
  ShowMessage(e.Message);
end;
```

نظرات خوانندگان

نویسنده: Alex's Blog
تاریخ: ۱۳۸۷/۱۰/۱۴ ۱۴:۴۴:۰۰

سلام. ممنون از مطالبتون. خیلی مفید هستند.
در ضمن قالبتون فکر کنم عوض شده مبارکه.

نویسنده: عرفان طاهری
تاریخ: ۱۳۸۷/۱۰/۱۴ ۲۱:۰۵:۰۰

به به قالب جدید مبارکه :D...

نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۰/۱۴ ۲۲:۵۴:۰۰

مرسی. این قالب قبلی که پیش فرض گوگل بود اصلا رنگ و رویی نداشت :)

نویسنده: مهدی
تاریخ: ۱۳۸۷/۱۰/۱۵ ۱۰:۵۰:۰۰

قالبه بهتره، ولی با اپرا، عنوان وبلاگ
(تازه‌های دنیای برنامه نویسی)به هم می ریزه.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۰/۱۵ ۱۱:۰۸:۰۰

فعلا با این مرورگرها تست شده:
IE7
فایرفاکس 3
کروم گوگل

نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۰/۱۵ ۱۲:۱۲:۰۰

با تشکر از دقت نظر شما. مشکل با opera هم برطرف شد.