MultiThreading**چيست؟**

مقدمه -

گاهي اوقات ممكن است كه شما بخواهيد برنامه شما دو يا چند عمل را به طور همزمان انجام دهد و يا اينكه نياز به انجام عملياتي كه مدت زمان زيادي به طول مي انجامد و يا زمان انجام آن معلوم نيست ، باشد ، بدون اينكه برنامه شما از دسترس كاربر خارج شود و به اصطلاح برنامه شما تا پايان يافتن عمليات قفل كند و همچنين كاربر بتواند عمليات را متوقف/معلق/شروع دوباره نمايد . در چنين موقعيتي نياز به MultiThreadingحس ميشود . به فرض مثال كد زير را در نظر بگيريد :

كد:

```
For i As Integer = 0 To 10000000
    For i2 As Integer = 0 To 100
        'Do Nothing
    Next
Next
```

هنگامي كه اين عمليات شروع ميشود ، كاربر توانايي كار با برنامه تا پايان يافتن آن را نخواهد داشت .

Thread**چيست؟**

Threadنامي براي جريان اجراي يک عمليات خاص ميباشد و هنگامي كه برنامه شما داراي چند Thread ميباشد بدان معناست كه قسمت هاي مختلفي از كد برنامه شما به طور همزمان در حال اجرا شدن ميباشند . در حقيقت كامپيوتر زمان پردازش يک عمليات را به قسمت (slice)هاي مختلفي تقسيم ميكند و هنگامي كه شما يک Thread جديد را آغاز ميكنيد كامپيوتر قسمتي از زمان را به آن اختصاص ميدهد . لازم به ذكر است كه برنامه شما از ابتدا داراي يک Thread اصلي (Main Thread) براي اجرا كد مربوط به آن ميباشد .

كار خود با Thread ها را آغاز مينماييم :

ميخواهيم برنامه اي بنويسيم كه تا يک عدد معين عمليات شمارش را انجام دهد .

1 – يک پروژه Windows Apploication به نام MutiThreading Sample ايجاد نماييد.

2 – يک Button به نام btnStart و يک TextBox به نام txtMAX به فرم اضافه نماييد .

3 – يک كلاس به نام clsCounter به پروژه اضافه كرده و كد زير را در داخل آن قرارهيد:

كد:

```
Public Class clsCounter
    Public MAX As Integer
```

```
    Public Event CountingFinished(ByVal Number As Integer)
    Sub StartCounting()
        Dim intTotal As Integer
        For i As Integer = 0 To MAX
            intTotal += 1
        Next
        RaiseEvent CountingFinished(intTotal)
    End Sub
End Class
```

توضيحات در مورد كد فوق :

•وظيفه اين كلاس شمردن از ١ تا مقدار MAX ميباشد .

•رويدادي با نام CountingFinished تعريف كرديم كه هنگامي كه عمليات شمارش به پايان برسد اتفاق مي افتد .

•متد StartCounting از ١ تا مقدار intMax را شماره كرده و در هر بار اجراي حلقه يک واحد به مقدار متغير intTotalاضافه ميشود كه در نهايت مساوي با مقدارMAX خواهد بود .

•پس از پايان شمارش رويداد CountingFinished را همراه با پاس كردن متغير intTotal به آن اجرا مينماييم .

حال ما بايد در هنگامي كه دكمه كليک ميشود يک Thread جديد ايجاد كرده و سپس متد StartCounting كلاس clsCounter را اجرا كرده و رخدادن رويداد CountingFinished را كنترل نماييم . در زماني كه عمليات شمارش انجام ميشود ما ميتوانيم رابط كاربري را كنترل كرده و كاربر توانايي كار با برنامه را دارد .

حال كد زير را به پروژه خود اضافه نماييد:

كد:

```
    Sub CountingFinishedEventHandler(ByVal N As Integer)
        System.Windows.Forms.MessageBox.Show("Counting Finished!")
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnStart.Click
        Dim CounterClass As New clsCounter
        Dim CountingThread As New Threading.Thread(AddressOf CounterClass.StartCounting)
        CounterClass.MAX = Val(txtMax.Text)
        AddHandler CounterClass.CountingFinished, AddressOf CountingFinishedEventHandler
```

```
    CountingThread.Start()
End Sub
```

توضيحات در مورد كد فوق :

•ابتدا يک پروسيجر براي كنترل رويداد CountingFinished مربوط به كلاس Counter ايجاد مينماييم .

هنگامي كه رويداد اتفاق بيافتد(عمليات شمارش به پايان برسد) ، پيغامي مبني بر پايان يافتن عمليات به كاربر نشان داده خواهد شد .

•در رويداد Click شي ء btnStart ، ابتدا يک نمونه از كلاس CounterClass ايجاد مينماييم .

•سپس براي ايجاد شي ء Thread ، آدرس متد clsCounter.StartCounting را به سازنده كلاس Thread پاس مينماييم به طوري كه متد clsCounter.StartCounting را بعد از آوردن كلمه كليدي addressof ، مي آوريم .

•بعد ، توسط كلمه كليدي Addhandle ، كنترل كننده رويداد كه CountingFinishedEventHandler نام دارد را به رويداد clsCounter.CountingFinished متصل مينماييم .

•در آخر نيز توسط متد Start مربوط به شيء CountingThread ، عمليات را آغاز مينماييم .

برخي متدهاي ديگر مربوط به شي ء : Thread

Suspendو : Resume در حالي كه يک Thread در حال اجراست ، توسط متد Suspend ميتوانيد آن را معلق كنيد كه منجر به متوقف شدن آن تا زماني كه متد Resume اجرا شود ، خواهد گرديد .

Abort : Threadرا متوقف ميكند .

Sleep : توسط اين متد ميتوانيد اجراي Thread را براي پاره اي از زمان (برحسب ميلي ثانيه) به حالت تعليق دربياوريد .

اولويت بندي Thread ها :

شما كنترل بيشتري بر رويThread ها داريد و ميتوانيد مقدار زماني كه هر Thread نسبت به ديگر Thread ها دريافت ميكند را از طريق خاصيت Priority تنظيم نماييد . اين خاصيت توسط يكي از ثابت هاي شمارشي زير كه عضوي از ThreadPriority ميباشد تنظيم ميشود:

ThreadPriority.AboveNormal : اولويت بالاتري به Thread ميدهد .

ThreadPriority.LowerPriority : اولويت پايين تري به Thread ميدهد .

ThreadPriority.HighestPriority : بالاترين اولويت را به Thread ميدهد .

ThreadPriority.LowestPriority : پايين ترين اولويت را به Thread ميدهد .

ThreadPriority.Normal : تولويت نرمال را به Thread ميدهد .

پيدا كردن وضعيت : Thread

وضعيت يک Thread را ميتوانيم به وسيله خاصيت ThreadState به دست بياوريم كه به وسيله يكي از

ثابتهاي شمارشي System.Threading.ThreadState معين ميگردد .

: System.Threading.ThreadState.Initializedبيان ميكند كه Thread مقداردهي اوليه شده اما هنوز شروع نگرديده است Thread : System.Threading.ThreadState.Ready>. آماده است .

: System.Threading.ThreadState.Runningبيان ميكند كه Thread در حال اجرا است .

: System.Threading.ThreadState.Standbyeبيان ميكند كه Thread در حالت آماده به كاراست .

: System.Threading.ThreadState.Initializedبيان ميكند كه Thread به پايان رسيده است .

: System.Threading.ThreadState.Transitionبيان ميكند كه Thread بين دو وضعيت بوده و در حالت انتقال از وضعيتي به وضعيت ديگر است .

: System.Threading.ThreadState.Unknownبيا ميكند كه وضعيت Thread معلوم نيست .

: System.Threading.ThreadState.Waitبيان ميكند كه Thread در حالت انتظار است.