

GÖRSEL PROGRAMLAMA

DR. ÖĞR. ÜYESİ HAYRİ VOLKAN AGUN
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ





İÇERİK

Vize %40, Final %60

Kitap:

Windows Forms Programming in C#, Chris Sells, AddisonWesley Professional; 2003; ISBN: 0321116208

Türkçe Kitap:

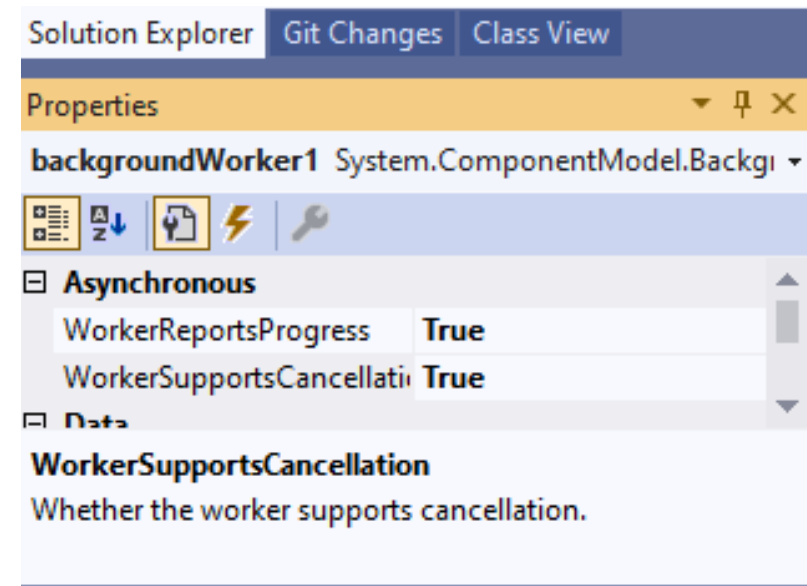
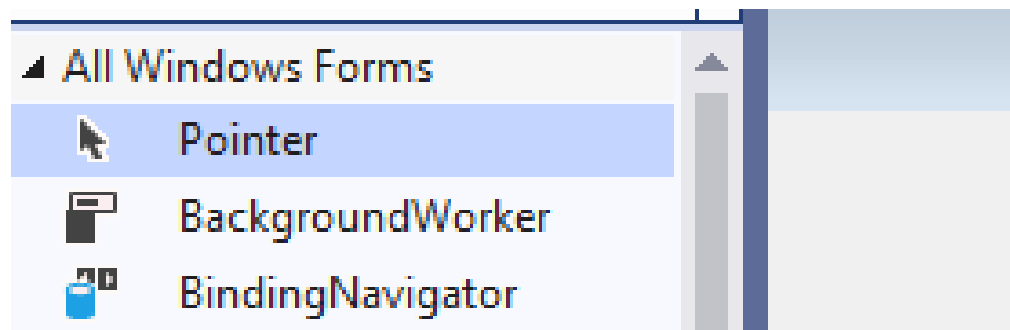
C# ile Görsel Programlama Eğitim Videolu, M. Furkan Ardoğan, Abaküs Kitap, 2020

Görsel Programlama - Background

- ❑ Arka planda sürekli çalışmasını istediğimiz kodlar için daha önce Thread örneğini görmüştük.
- ❑ Ancak Thread örneğinde görsel ara yüzdeki bileşenler için güncelleme yaparken sorun yaşanabilir. Örneğin tam Thread bir güncelleme yapacakken kullanıcı bileşen üzerinde örneğin klavyeden bir tuşa basıp yazı yazarsa işlem kesintiye uğrayabilir.
- ❑ Bu tür durumlar asenkron (asynchronous) işlem denir. Kısaca oluşan olaylar çakışabilir.
- ❑ Bunun dışında diğer bir problemde görsel bileşenlerin birden fazla Thread ile erişiminde yaşanabilecek çok yoğun çalışma durumuna uygun olmamalarıdır.
- ❑ Eğer çok yoğun bir Thread erişimi ile mevcut görsel UI bileşen sürekli değişiyorsa bu durumda ekrana çizme işlemi bile gecikecek ve bileşen kullanıcı erişimine kayıtsız kalacaktır.

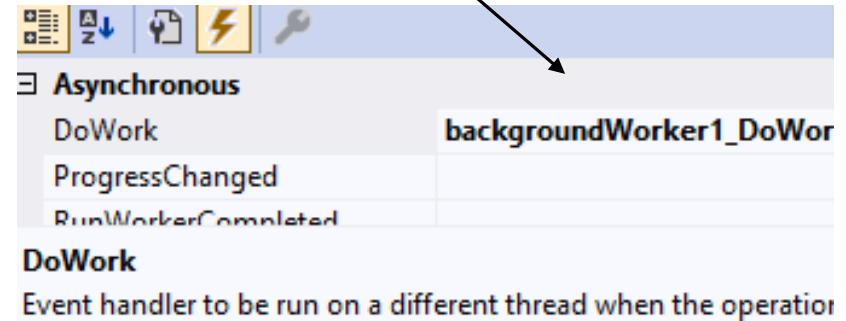
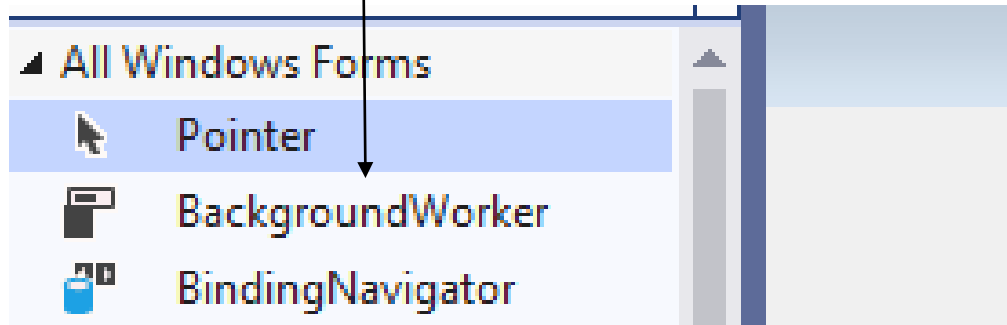
Görsel Programlama - Background

- ❑ Tüm bu durumları engellemek Windows forms içerisinde BackgroundWorker adında bir bileşen tanımlanmıştır.
- ❑ BackgroundWorker bir Thread yapısıdır ve bileşen güncellemelerini arka planda senkron olacak şekilde yapabilmemize olanak tanır.
- ❑ WorkerReportProgress ve WorkerSupportsCancellation True olarak seçilir.



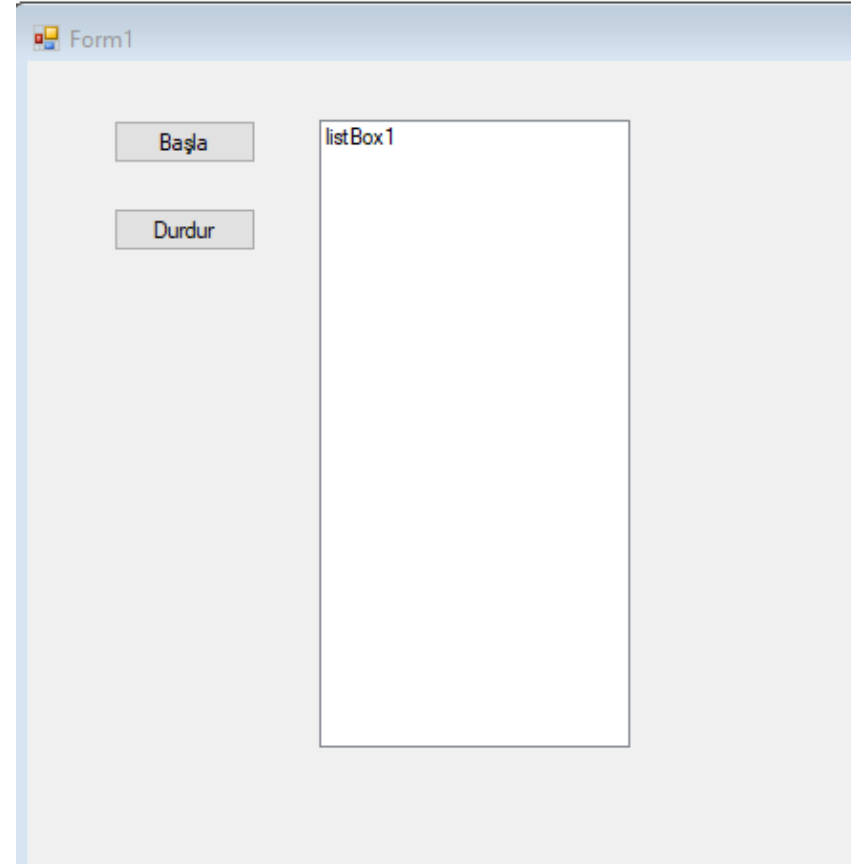
Görsel Programlama - Background

- ❑ BackgroundWorker için BackgroundWorker bileşeninin DoWork event'i kullanılarak arka planda çalışacak paralel işlem/kod fonksiyonu yazılır.



Görsel Programlama - Background

□ Tasarımda başlama ve durdurma işlemleri için Başlama butonu ve Durdurma butonu eklenmiştir. Bu işlemler ile ListBox bileşeni içerisine arka planda rastgele sayılar eklenmeye başlayacak veya bu işlem durdurulacaktır.



Görsel Programlama - Background

- ❑ Kod ekranında DoWork eventi içinde yeni bir sayı üretilerek ListBox1 içerisine eklenmek üzere kaydedilir.

devamEt ile Durdur butonuna basarak döngüden çıkmamız sağlanır.

```
private void backgroundWorker1_DoWork(object sender, DoWorkEventArgs e)
{
    while (devamEt)
    {
        Random rnd = new Random();
        value = rnd.Next();
        Thread.Sleep(1000);
        backgroundWorker1.ReportProgress(0, value.ToString());
    }
}
```

ReportProgress kullanılarak değişen değer kaydedilir.

Görsel Programlama - Background

ProgressChanged event'i kullanılarak
kullanılarak kaydedilen değer listBox1
içerisine eklenerek bileşen güncellenir.

1 reference

```
private void backgroundWorker1_ProgressChanged(object sender, ProgressChangedEventArgs e)
{
    listBox1.Items.Add(sender.ToString());
}
```


Görsel Programlama - Background

Buton1 click even'i kullanılarak önce backgroundworker başlatılır. Sonrada Buton2 click even'i ile durdurulur.

```
1 reference  
private void button1_Click(object sender, EventArgs e)  
{  
    backgroundWorker1.RunWorkerAsync();  
}
```

```
1 reference  
private void button2_Click(object sender, EventArgs e)  
{  
    backgroundWorker1.CancelAsync();  
    devamEt = false;  
}
```