****

**ÇANKIRI KARATEKİN ÜNİVERSİTESİ MESLEK YÜKSEKOKULU**

**BİLGİSAYAR TEKNOLOJİLERİ BÖLÜMÜ**

**BİLGİSAYAR PROGRAMCILIĞI PROGRAMI**

**CBL-205 SİSTEM ANALİZİ VE TASARIMI I DERSİ**

**ÖDEV RAPORU**

**ÖDEVİN ADI: C# DELEGELER , EVENTLER(TEMSİLCİ KULLANIMI VE OLAYLAR ) VE ASENKRON PROGRAMLAMA**

**DERS SORUMLUSU: ÖGR. GÖR. İSMAİL KARAMAN**

**HAZIRLAYAN**

**220102068-EREN KARABIYIK-2.SINIF/B Şubesi**

**Çankırı 2024**

**İÇİNDEKİLER**

* **Delege nedir? Hangi amaçla kullanılır?**
* **Event Nedir? C# da Event nasıl tanımlanır. Bir eventi handle etmek ne anlama gelir?**
* **Asenkron programlama nedir? Hangi durumlarda asenkron programlama yapma ihtiyacı duyarız?**

**Delge nedir ? Hangi amaçla kullanılır ?**

Delegenin sözlük anlamı kendisine devleti ya da halkı ya da bir kuruluşu, kurumu temsil yetkisi verilerek bir yere, bir toplantıya gönderilen kimsedir . Yani aslında elçi diyebiliriz . Proglama dillerinde ise hangi sınıftan hangi metodu çağıracağımızı bilmeden, çağrılacak metodun neye benzediğini belirlememizi sağlar.

C# dilinde delegelerinkullanım amacı metot saklamaktır. DelegelerOlay(event) tabanlı programlama ve asenkron programlama yaparken, anonim metot yazarken kullanılır. Bir diğer kullanım amacı da, bir metoda parametre olarak başka bir metot verebilmektir.

**Event (olay) nedir ?**

Programlamada "event" (olay) terimi, bir programın çalışması sırasında meydana gelen belirli bir durumu veya kullanıcının gerçekleştirdiği bir eylemi ifade eder. Bu olaylar, genellikle bir programın çalışma zamanında gerçekleşen belirli durumları veya kullanıcının etkileşimini temsil eder. Programlama bağlamında, olaylar genellikle kullanıcı etkileşimleri, ağ olayları ve diğer bir dizi durumu içerir. Olaylar, genellikle bir programın belirli bir parçasında (bir fonksiyon, metod veya olay işleyici) belirtilen bir tepkiyi tetikler. Birkaç programlama paradigmada olay tabanlı programlama kullanılır.

C# programlama dilinde olaylar (events) genellikle bir delegeden türetilir. Olaylar, genellikle bir nesnenin belirli bir durumu veya etkileşimi temsil eder.

**Peki bir eventi handle etmek nedir ?**

Bir eventi handle etmek (ele almak), programlama bağlamında bir olayın gerçekleştiğinde bu olaya tepki göstermek veya işlemek anlamına gelir. Event handling (olay ele alma), genellikle kullanıcı etkileşimi veya başka bir nesnenin durumu gibi bir dizi olaya tepki göstermek amacıyla kullanılır. Örneğin bir kullanıcının bir butona tıklaması , bir sensörün belirli bir değeri algılaması gibi durumlar olayları tetikleyebilir. Event handling, genellikle şu adımları içerir .

**Olayın Tanımlanması (Event Definition) :** Olay , bir sınıf içinde bir delegeden türetilerek veya özel bir olay tanımıyla belirtilir.

**Olaya Abonelik (Event Subscription) :** Olaya tepki göstermek isteyen diğer kod blokları , olaya abone olurlar. Bu, genellikle bir fonksiyon , metod veya olay işleyiciyi olaya bağlama işlemidir.

**Olayın Tetiklenmesi (Event Triggering) :** Belirli bir durum gerçekleştiğinde (örneğin, bir butona tıklama), olay tetiklenir.

**Olayın Ele Alınması (Event Handling) :** Olaya abone olan kod bloğu (event handler) olay gerçekleştiğinde çalıştırılacak kodu içerir. Bu kod, genellikle belirli bir işlevi yerine getirir veya olayın gerçekleştiği duruma göre belirli bir tepkiyi sağlar.

**Asenkron programlama nedir ?**

Asenkron proglama işin parçalara ayrılıp tüm işleri birarada yapılmasını sağlar . Asenkron programlama ile program içeriside yazdığımız kodlar işletilirken aynı programa içerisinde diğer kodlarda işletilebilir . Bu sayede bir bölüm kullanılırken diğer bir bölüm de işlem yabılabilir .

Asenkron proglama multi threading ile karıştırılmamalıdır . Asenkron olarak yazacağımız kodlar tek bir thread üzerinde çalışabilir .

Peki multi threading nedir ? Buna değinecek olursak multi threading (çoklu iş parçacığı) bir bilgisayar programının aynı anda birden fazla iş parçacığında çalışmasını ifade eden bir konsepttir. İş parçacığı (thread), bir program içinde bağımsız olarak yürütülebilen en küçük yürütme birimidir. Multi threading bilgisayar kaynaklarını daha etkili bir şekilde kullanma paralel işleme sağlama ve programların daha hızlı tepki vermesini sağlama amacı taşır.

**Son olarak hangi durumlarda Hangi durumlarda asenkron programlama yapma ihtiyacı duyarız?**

Asenkron programlama, belirli durumlarda daha etkili ve verimli bir program tasarlamak için kullanılır. İşte asenkron programlamaya duyulan ihtiyaç duyulan bazı durumlar

**Uzun Süren İşlemler :** Eğer bir programda uzun süren işlemler (örneğin, ağ üzerinden veri indirme, dosya okuma/yazma, veritabanı işlemleri) varsa bu işlemleri asenkron olarak yürütmek, programın diğer kısımlarının bu süreçlerin tamamlanmasını beklemeksizin devam etmesine olanak tanır. Bu, programın daha hızlı ve daha duyarlı olmasını sağlar.

**Kullanıcı Arayüzü Tepkileri :** Grafik kullanıcı arayüzleri (GUI) kullanılan uygulamalarda kullanıcı ile etkileşim sırasında programın hemen cevap vermesi önemlidir. Asenkron programlama, kullanıcı etkileşimlerini ve GUI olaylarını etkileşimli ve daha hızlı bir şekilde ele almayı sağlar.

**Paralel İşlemler ve Çoklu Çekirdekli Sistemler :** Çoklu çekirdekli bilgisayar sistemleri artık yaygındır. Asenkron programlama, bu çekirdeklerde paralel olarak çalışabilen iş parçacıkları veya görevler oluşturmayı kolaylaştırır, bu da genellikle performans artışına yol açar.

**Web Geliştirmesi ve İstemci-Tarayıcı İletişimi :** Web uygulamalarında, tarayıcı ile sunucu arasındaki iletişim genellikle asenkron teknikler kullanılarak yapılır. Örneğin, AJAX (Asenkron JavaScript ve XML), web sayfalarının güncellenmesini sayfa yenilemeye gerek kalmadan gerçekleştirmek için kullanılır.

**Zamanlamaya Bağlı İşlemler :** Belirli bir zaman aralığında tekrarlanan görevler veya belirli bir zamanın geçmesini bekleyen işlemler, asenkron programlamayla daha etkili bir şekilde yönetilebilir.

**Etkinlik Tabanlı Programlama :** Olay tabanlı programlamada, bir program belirli olaylara (events) tepki verir. Asenkron programlama, bu tür etkinliklere dinamik bir şekilde ve hızlı bir yanıtla tepki verme yeteneği sağlar .