



ALICAN YÜCEL

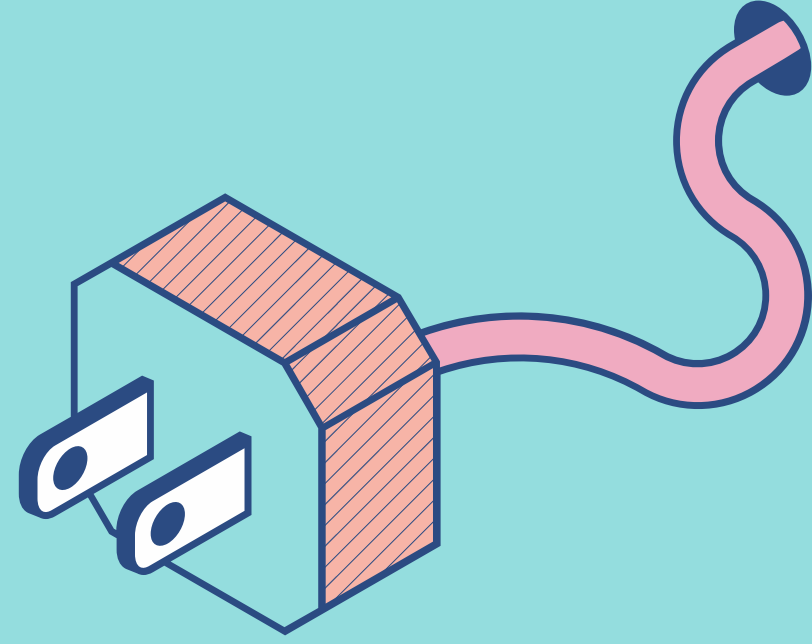
# C# 12 ile Gelen Yenilikler

BTK AKADEMİ

# C# 12 İle Gelen Yenilikler

C# 12 ve .NET 8 geliştiricilere performans ve verimlilik açısından önemli avantajlar sunuyor. Bu yeni özellikler, gelecekteki performans iyileştirmelerinin temellerini oluşturuyor ve yazılım geliştirmeyi daha kolay ve verimli hale getiriyor.

# Primary Constructors



C#'ın 12. versiyonu, sınıf ve yapılar içinde temel yapılandırıcılar (primary constructors) oluşturmayı mümkün kılıyor. Artık temel yapılandırıcılar sadece kayıt tiplerle sınırlı değil. Temel yapılandırıcı parametreleri, sınıfın tüm gövdesi boyunca erişilebilir hale geliyor. Tüm açıkça bildirilen yapılandırıcılar, `this()` sözdizimini kullanarak temel yapılandırıcıyı çağırmalıdır, böylece tüm temel yapılandırıcı parametreleri kesin olarak atanır. Bir sınıfa temel yapılandırıcı eklemek, derleyicinin otomatik olarak parametresiz bir yapılandırıcı oluşturmasını engeller. Yapıda, otomatik olarak oluşturulan parametresiz yapılandırıcı, tüm alanları, temel yapılandırıcı parametrelerini de içerecek şekilde 0-bit deseniyle başlatır.

Birincil yapılandırıcı parametreleri için sadece kayıt tiplerde (record class veya record struct tipleri) derleyici tarafından otomatik olarak genel özellikler (public properties) oluşturulur. Kayıt olmayan sınıf ve yapılar, birincil yapılandırıcı parametreleri için her zaman bu davranışı istemeyebilir.

# Inline Diziler: Performansı Artıran Yenilik

`InlineArrayAttribute`, daha önceki bir .NET 8 önizlemede tanıtılan önemli bir özelliktir. Bu, C#'ın 12. versiyonunda daha da geliştirilerek performansı artıran bir araç haline gelmiştir. Inline diziler, bir türün sürekli bir dizi temsilini tanımlamak için kullanılır ve aşırı taşmaları engellemek için indekslenmesi ve kesilmesi kolaydır. Bu tür veriler, etkin ve hızlı bir şekilde işlenir ve .NET kitaplıkları ve bazı diğer kütüphaneler tarafından yaygın olarak kullanılır.

```
private static void InlineArrayAccess(Buffer10<int>  
    inlineArray) { for (int i = 0; i < 10; i  
    ++){ inlineArray[i] = i * i; } foreach (int i in  
    inlineArray) { Console.WriteLine(i); } }
```

İBurada Buffer10 adında bir yapı (struct) kullanılarak inline bir dizi oluşturuyoruz. Bu yapı, InlineArrayAttribute ile işaretlenmiştir ve 10 elemanlı bir dizi temsil eder. İlk 10 sayının karesini hesaplayarak ve yazdırarak bu inline diziyi kullanıyoruz.

# Interceptors: Yenilikçi ve Deneysel Bir Yaklaşım

C#'ın 12. versiyonuyla birlikte deneysel bir özellik olan interceptors tanıtıldı. Bu özellik, derleme öncesi (AOT) derleme için daha iyi destek sağlayan gelişmiş senaryolar için tasarlanmıştır. Interceptors, özel kodun belirli yöntem çağrılarını yönlendirmesine olanak tanır. Özellikle kaynak üreticileri için uygundur ve belirli kodları yeniden yönlendirme yeteneği sunar.

```
[Features("InterceptorsPreview")] public class  
    MyService { [Interceptor(typeof(MyInterceptor))] public void  
        DoSomething() { // Burada yapılacak işlem } }
```

Yukarıdaki örnek, MyService sınıfının DoSomething() yönteminin MyInterceptor adlı özel bir interceptor ile yeniden yönlendirilmesini göstermektedir.



# Nameof ile Erişilebilen Öğeler

iC#'ın 12. versiyonuyla birlikte nameof anahtar kelimesi, artık üye adları, başlatıcılar, statik üyeler ve niteliklerle çalışır. Bu özellik, C# dilinin daha esnek ve kullanıcı dostu olmasını sağlar. Özellikle, adları bilinen üyeleri ve nitelikleri kullanırken, daha az hata yapma olasılığını artırır.

```
internal class NameOf
{
    public string S { get; } = "";
    public static int StaticField;
    public string NameOfLength { get; } = nameof(S.Length);
    public static void NameOfExamples()
    {
        Console.WriteLine(nameof(S.Length));
        Console.WriteLine(nameof(StaticField.MinValue));
    }
    [Description($"String {nameof(S.Length)}")]
    public int StringLength(string s)
    {
        return s.Length;
    }
}
```

Yukarıdaki örnekte, nameof anahtar kelimesi, sınıfın içindeki öğelerin adlarına ve niteliklerine erişim sağlamak için kullanılır.

# Varsayılan Lambda Parametreleri ve Tüm Tiplere Takma Ad Verme

Bu yeni versiyon ile gelen iki önemli özellik, yazılım geliştirme deneyimini daha zenginleştiriyor.

**Varsayılan Lambda Parametreleri:** Artık lambda ifadelerinde parametreler için varsayılan değerler belirleyebilirsiniz. Sözdizimi ve kurallar, herhangi bir yöntem veya yerel fonksiyon için argümanlara varsayılan değer eklemekle aynıdır. Bu, kodunuzu daha esnek ve okunabilir hale getirirken, lambda ifadeleri ile çalışmayı daha kolay ve verimli hale getirir.

**Tüm Tiplere Takma Ad Verme:** `using alias` direktifi, artık yalnızca adlandırılmış tipler için değil, tüm tipler için takma adlar oluşturmanıza olanak tanır. Bu, tuple tipleri, dizi tipleri, işaretçi tipleri veya diğer güvensiz tipler için anlamsal takma adlar oluşturmanızı sağlar. Bu sayede, kodunuzun daha anlaşılır ve temiz olmasını sağlayabilirsiniz.

| Özellik                                    | Preview Sürümü               |
|--|------------------------------|
| Primary constructors                       | Visual Studio 17.6 preview 2 |
| Lambda ifadelerinde opsiyonel parametreler | Visual Studio 17.5 preview 2 |
| Tüm tiplere takma ad verme                 | Visual Studio 17.6 preview 3 |
| Inline diziler                             | Visual Studio 17.7 preview 3 |
| Interceptorlar (Preview)                   | Visual Studio 17.7 preview 3 |

# SONUÇ

- C#'ın 12. versiyonu ve .NET 8'in önizlemesi, yazılım geliştirmeye önemli katkılarda bulunuyor. Inline diziler sayesinde performans artışı elde ederken, interceptors ile daha yenilikçi kod yapıları oluşturmak mümkün oluyor. Ayrıca nameof ile daha güvenli ve esnek bir kodlama deneyimi yaşıyor. Tüm bu özellikler, gelecekteki performans iyileştirmelerine zemin hazırlıyor ve geliştiricilere daha etkili araçlar sunuyor.
- Bu yenilikçi özelliklerle C# 12 ve .NET 8, yazılım dünyasında yeni bir döneme işaret ediyor. Daha hızlı ve verimli uygulamalar geliştirmek için Visual Studio önizleme sürümünü veya .NET SDK'nın en son sürümünü indirebilir ve projenizin dil sürümünü önizlemeye ayarlayabilirsiniz.
- Daha fazla bilgi için Microsoft Learn'daki "C# 12'de Neler Yeni" sayfasına ve Roslyn Özellik Durumu sayfasına göz atabilirsiniz. Siz de bu yenilikçi özellikleri deneyerek yazılım geliştirme deneyiminizi geliştirebilirsiniz.

