

# C# Eğitim

## KONU

### Değişkenler

#### GENEL BAKIŞ

- ❖ **Değişkenler**, bir programlama dilinde değerlerin depolanabileceği bellek lokasyonlarıdır. Değişkenler, programın çalışması sırasında değiştirilebilir ve farklı değerlerle yeniden atanabilir. Bu nedenle, değişken adı, içinde depolanan değer türüne ve anlamına uygun olarak seçilir. Değişkenler, matematiksel ifadelerde kullanılabilir veya programda belirli bir işlevi yerine getirmek için kullanılabilir.
- ❖ Programlamada değişkenler, bellek yönetimini etkileyen önemli bir faktördür. Bellek, bilgisayarın geçici olarak verileri depoladığı yerdir ve programlar bellek üzerinde çalışır. Değişkenler, programın bellek kullanımını etkileyebilir ve aynı zamanda programın doğru çalışması için gereksiz bellek kullanımını da engelleyebilir.
- ❖ Değişkenler, programlama dillerinde genellikle temel veri tipleri (örneğin, sayılar veya metinler) veya nesneler gibi karmaşık veri yapıları gibi farklı türlerde tanımlanabilir. Bu veri tipleri, değişkenin bellekte ne kadar yer kaplayacağını ve hangi işlemlerin yapılacağını belirler.

DEĞİŞKENLER	İÇERİK	ÖRNEK	TÜRKÇE KAYNAK
DEĞİŞKEN TANIMLAMASI	<ul style="list-style-type: none"><li>❖ Değişkenleri tanımlamak, değişkenin adını ve türünü belirlemek için kullanılan sözdizimini anlamak önemlidir.</li></ul>	<a href="#">Link1-C#Station</a> <a href="#">Link2-TutorialsPoint</a> <a href="#">Link3-GeeksForGeeks</a> <a href="#">Link4-CSharpCorner</a> <a href="#">Link5-LearnCs</a>	<a href="#">Link1-MicrosoftC#</a> <a href="#">Link2-azKod</a> <a href="#">Link3-MuratYdağ</a>
VERİ TİPLERİ	<ul style="list-style-type: none"><li>❖ Programlama dillerinde farklı veri tipleri vardır ve her biri belirli bir amaç için kullanılır.</li></ul>	<a href="#">Link1-Microsoft</a> <a href="#">Link2-W3Schools</a> <a href="#">Link3-TutorialsPoint</a>	<a href="#">Veri Tipleri ve Değişken Tanımlama Yazılım Kodlama</a>
ATAMA İŞLEMİ	<ul style="list-style-type: none"><li>❖ Değişkenlere değer atamak, bir değişkenin bellekte ne kadar yer kaplayacağını belirleyebilir. Atama işlemi sırasında, değişkenin ayrılması gereken bellek miktarına karar vermek önemlidir.</li></ul>	<a href="#">Link1-W3Schools</a> <a href="#">Link2-Microsoft</a> <a href="#">Link3-TutorialsPoint</a>	<a href="#">Link1</a> <a href="#">Link2</a> <a href="#">Link3</a>

**DEĞİŞKENLER****İÇERİK****ÖRNEK****TÜRKÇE KAYNAK**

<b>DEĞİŞKEN KULLANIMI</b>	❖ Değişkenler, programda bilgi depolamanın yanı sıra, ifadelerde, koşullu ifadelerde ve döngülerde de kullanılır.	<a href="#">Link1-Microsoft</a> <a href="#">Link2-GeekforGeeks</a> <a href="#">Link3-CodeAcademy</a>	<a href="#">Link1</a> <a href="#">Link2</a> <a href="#">Link3</a>
<b>DEĞİŞKEN KAPSAMI</b>	❖ Değişkenlerin kapsamı, bir değişkenin hangi bölgede geçerli olduğunu belirler. Yerel veya küresel değişkenler olabilir ve bu durumlar, değişkenin kapsamını belirleyen programlama diline bağlıdır.	<a href="#">Link1-C#Station</a> <a href="#">Link2-GeeksForG</a>	
<b>BELLEK YÖNETİMİ</b>	❖ Değişkenlerin bellek yönetimi, programların performansını etkileyebilir.		

**VERİ TÜRLERİ KAYNAK****İÇERİK****ÖRNEK****TÜRKÇE**

Sayısal Veri Türleri	❖ Sayılar, programlama dillerinde sıklıkla kullanılır ve tam sayılar (integer) veya ondalık sayılar (float) gibi farklı sayısal veri türleri vardır.	<a href="#">Link1-Microsoft</a> <a href="#">Link2-TutorialsT</a> <a href="#">Link3-TutorialsP</a> <a href="#">Link4-CodeAcademy</a>	<a href="#">Link1-Microsoft</a>
Karakter Dizisi Veri Türleri	❖ Metinleri depolamak için karakter dizisi (string) veri türü kullanılır.	<a href="#">Link1-Microsoft</a> <a href="#">Link2-Tutorials</a>	
Boolean Veri Türü	❖ Boolean veri türü, yalnızca iki olası değere sahiptir: doğru (true) ve yanlış (false).	<a href="#">Link1-Microsoft</a>	

Listeler	❖ Liste, programlamada birden fazla öğeyi depolamak için kullanılan bir veri türüdür. Listelerin elemanları, sabit bir şekilde sıralanmıştır ve her elemanın kendi veri türüne sahip olması mümkündür.	<a href="#">Link1-Microsoft</a> <a href="#">Link2-DotnetP</a> <a href="#">Link3-TutorialsT</a>	
Sözlük	❖ Sözlükler, anahtar-değer çiftleri olarak depolama sağlarlar ve bir anahtarın karşılık geldiği değere erişmek için kullanılırlar.	<a href="#">Link1-Microsoft</a>	
Nesneler	❖ Nesneler, programlama dillerinde daha karmaşık veri yapıları oluşturmak için kullanılırlar. Nesneler, özellikleri (property) ve yöntemleri (method) içerebilir.	<a href="#">Link1-Microsoft</a> <a href="#">Link2-DotnetP</a>	

## KONU

### Şart İfadeleri

#### GENEL BAKIŞ

- ❖ C# şart ifadeleri, belirli koşulların sağlanması durumunda belirli bir kod bloğunun çalıştırılmasını sağlayan yapılar olarak tanımlanabilir. Bu yapılar, C# programlama dilinde kontrol yapıları arasında yer alırlar ve programın akışını belirlemek için kullanılırlar.
- ❖ C# şart ifadeleri, genellikle if-else veya switch-case yapısı kullanılarak oluşturulur. Bu yapılar, belirli bir koşulun doğru veya yanlış olması durumuna göre programın farklı bir yol izlemesini sağlar. Örneğin, if-else ifadesi, belirli bir koşulun doğru olması durumunda belirli bir kod bloğunu çalıştırırken, aksi takdirde farklı bir kod bloğunu çalıştırır.
- ❖ Ayrıca, C# programlama dilinde ternary operatörü olarak bilinen kısa bir şart ifadesi de vardır. Bu yapı, bir koşulun doğru veya yanlış olması durumuna göre farklı bir değeri döndürür.

[Link1-W3Schools](#)

[Link2-Microsoft](#)

[Link3-Learncs](#)

## KONU

### Döngüler

#### GENEL BAKIŞ

- ❖ C# döngüler, belirli bir işlem veya kod bloğunun tekrar edilmesini sağlayan bir yapıdır. Döngüler, belirli bir koşul doğru olduğu sürece tekrar eden bir dizi işlemi gerçekleştirirler.
- ❖ C# programlama dilinde, üç farklı döngü türü vardır:
- ❖ for döngüsü: Belirli bir sayıda tekrar için kullanılır. For döngüsü, koşullu bir değişkenin belirli bir sayıda arttırılması veya azaltılması için kullanılır.
- ❖ while döngüsü: Belirli bir koşul doğru olduğu sürece tekrar eder. While döngüsü, belirli bir işlemi, koşul doğru olduğu sürece tekrar eder.
- ❖ do-while döngüsü: Do-while döngüsü, koşul doğru olduğu sürece belirli bir işlemi tekrar eder. Ancak, koşul doğru olmadığı zaman bile bir kere daha işlem yapar.
- ❖ Döngüler, programlama dilinde tekrar eden işlemlerin çoğunu yürütmek için kullanılır. Örneğin, bir liste elemanlarını işlemek veya bir dosyadaki satırları okumak için döngüler kullanılabilir.

[Link1-Microsoft](#)

[Link2-TutorialsPoint](#)

[Link3-Csharp-Station](#)

[Link4-W3Schools](#)

## KONU

### Diziler

#### GENEL BAKIŞ

- ❖ C# programlama dilinde, diziler, aynı veri tipindeki öğelerin tek bir değişken adı altında depolanmasına izin veren bir veri yapısıdır. Bu öğeler, sıralı bir şekilde saklanır ve bir dizideki her öğe, bir dizinin dizin numarası veya indeksi adı verilen bir tam sayıyla belirtilir.
- ❖ Diziler, birçok benzer türdeki verileri saklamak için kullanılabilir. Örneğin, bir dizi içinde birkaç tam sayı, birkaç ondalık sayı veya hatta birkaç dize saklayabilirsiniz.
- ❖ Dizilerin boyutu önceden belirlenmiş veya değişken olabilir. Dizi boyutu önceden belirlenmişse, o dizi için ayrılan bellek bloğu, tanımlandığında ayrılmıştır ve program çalıştırıldığında değiştirilemez. Öte yandan, boyutu değişken olan bir dizi, program çalıştırıldığında dinamik olarak oluşturulabilir.
- ❖ C# programlama dilinde, bir dizi oluşturmak için **[]** sembolleri kullanılır.

[Microsoft](#)

[Geeksforgeeks](#)

[TutorialsPoint](#)

[Guru99](#)

## KONU

### Fonksiyonlar

#### GENEL BAKIŞ

- ❖ C# programlama dilinde fonksiyonlar, kod blokları içinde belirli bir işlevi yerine getiren ve belirli bir isimle çağrılabilen kod bloklardır. C# dilinde fonksiyonlar metot olarak da adlandırılır. Fonksiyonlar, aynı kodu birkaç kez tekrarlamaktan kaçınmak, daha düzenli kod yazmak, daha az hata yapmak, kolayca değiştirilebilir kod yazmak gibi birçok fayda sağlar.
- ❖ C# dilinde fonksiyonlar, iki ana kategoriye ayrılır: void ve değer döndüren fonksiyonlar.
- ❖ void fonksiyonlar: Geriye herhangi bir değer döndürmez. Sadece belirli bir işlevi yerine getirir ve işlem sonunda bir değer döndürmez. Void fonksiyonlar genellikle bir işlem yapmak için kullanılır.
- ❖ Değer döndüren fonksiyonlar: Geriye belirli bir değer döndürür. Bu değer, fonksiyon tarafından hesaplanan bir sayı, bir karakter veya bir dizi de olabilir. Değer döndüren fonksiyonlar genellikle bir işlem yapar ve sonucu döndürür.
- ❖ Örnek olarak, C# dilinde "Main" fonksiyonu, bir C# uygulamasının çalışması için gereklidir. Bu fonksiyon, C# kodu tarafından otomatik olarak çağrılır ve programın ana işlevi olarak kabul edilir.

[Microsoft](#)

[TutorialsPoint](#)

[GeeksforGeeks](#)

## KONU

### Nesne Yönelimli Programlama (OOP)

#### GENEL BAKIŞ

- ❖ Nesne yönelimli programlama (Object-Oriented Programming- OOP), programlamada kullanılan bir paradigmadır. Bu paradigma, programlamada kullanılan verileri (değişkenler) ve bu veriler üzerinde işlem yapan işlevleri (fonksiyonlar) bir araya getirerek, bir nesne olarak ele alır.
- ❖ OOP'de programlama yaparken, gerçek dünyadaki nesnelerin yapısını taklit etmek için sınıflar kullanılır. Sınıflar, belirli özelliklere ve işlevlere sahip nesneleri tanımlamak için kullanılır. Örneğin, bir araba sınıfı, araba nesnelerinin özelliklerini (renk, marka, model, vb.) ve işlevlerini (sürüş, durma, vb.) tanımlayabilir.
- ❖ OOP'nin en önemli özelliklerinden biri, nesne tabanlı programlamada kullanılan kalıtım ve çok biçimlilik gibi özelliklerdir. Kalıtım, bir sınıfın, başka bir sınıfın özelliklerini miras almasına olanak tanıırken, çok biçimlilik, bir nesnenin farklı biçimlerde davranabilmesine olanak tanır.
- ❖ OOP, büyük ve karmaşık programları, daha modüler ve anlaşılır hale getirir. Bu nedenle, birçok modern programlama dilinde, nesne yönelimli programlama özelliği bulunmaktadır. C# da nesne yönelimli programlama özelliklerini destekleyen bir programlama dilidir.

[Microsoft](#)

[CSharpCorner](#)

[Medium](#)

[Patika](#)

[Turkcell](#)

[TalentGrid](#)

## KONU

### İleri Seviye Nesne Yönelimli Programlama (OOP)

#### GENEL BAKIŞ

- ❖ İleri seviye nesne yönelimli programlama, nesne yönelimli programlama (OOP) kavramlarını daha da derinleştiren ve genişleten bir programlama yaklaşımıdır. Bu yaklaşım, özellikle büyük ölçekli ve karmaşık yazılımların geliştirilmesinde yararlı olabilir.
- ❖ İleri seviye nesne yönelimli programlama, özellikle polymorphism (çok biçimlilik), inheritance (kalıtım) ve encapsulation (kapsülleme) gibi OOP prensiplerinin daha karmaşık uygulamalarında kullanılan teknikleri kapsar. Ayrıca, design patterns (tasarım kalıpları) gibi genel programlama prensipleri ve teknikleri de ileri seviye nesne yönelimli programlama içinde yer alır.
- ❖ Bu yaklaşım genellikle büyük ölçekli projelerin geliştirilmesinde kullanılır ve kodun daha okunaklı, yönetilebilir ve sürdürülebilir olmasını sağlar.
- ❖ C# gibi modern programlama dilleri, ileri seviye nesne yönelimli programlamayı destekler ve bu konuda birçok kaynak ve araç mevcuttur.

[Microsoft](#)

## KONU

### SQL SERVER

#### GENEL BAKIŞ

- ❖ SQL Server, Microsoft tarafından geliştirilen ve yönetim veritabanı sistemlerinde kullanılan bir ilişkisel veritabanı yönetim sistemidir (RDBMS). SQL Server, kullanıcıların veri yönetimini ve depolamasını sağlayarak, veri tabanlarına erişmek ve veri yönetmek için SQL dilini kullanır.
- ❖ SQL Server, yüksek performans, güvenlik, işlevsellik ve ölçeklenebilirlik sağlamak için tasarlanmıştır. Ayrıca işletmelerin ihtiyaç duyduğu veri depolama, işlem yönetimi, iş zekası ve raporlama gibi işlevleri yerine getirir. SQL Server, Windows işletim sistemi üzerinde çalışır ve Microsoft'un diğer yazılımları ile entegre edilebilir.

[W3Schools](#)

[sqltutorial](#)

## Konu Anlatım Videoları

[Yazılım Teknolojileri Akademisi](#) – C#

[Engin Demiroğ](#) – C#

[Enes Bayram](#) -C#

[Yazılım Teknolojileri Akademi](#)- SQL

### Örnek Sorular

<https://www.azkod.com/csharp/c-basit-konsol-ornekleri>

<https://www.evrenselkod.com/c-sharp-ornekleri/>

<https://www.mitopya.com/c-ornekleri-basitten-zora-console-ornekleri/>

<https://www.algoritmaornekleri.com/c-sharp/console/c-kod-ornekleri/>

<https://www.yazilimkodlama.com/programlama/c-basit-console-ornekleri/>

<https://www.evrenselkod.com/c-sharp-test-sorulari/>

<https://www.gencayyildiz.com/blog/c-algoritma-sorulari-ve-cozumleri/>