

## Programlama Dillerini Seviyelere Göre Sınıflandırma

Programlama dilleri, genel olarak donanıma olan yakınlıkları ve soyutlama seviyeleri bazında üç ana kategoriye ayrılır:

### 1. Düşük Seviyeli Diller

- **Makine Dili:** Bilgisayarın doğrudan anlayabildiği 0 ve 1'lerden oluşan en temel dildir. Çok hızlıdır ancak anlaşılması ve yazılması zordur.
- **Assembly Dili:** Makine dilinin sembolik bir temsilidir. Makine diline çok yakın olduğu için performanslıdır ancak hala düşük seviyeli bir dildir.

#### Özellikleri:

- Donanıma çok yakın çalışırlar.
- Bellek yönetimi ve işlemci kontrolü gibi düşük seviyeli işlemleri doğrudan yapmaya izin verirler.
- Hızlı ve verimlidirler.
- Anlaşılması ve yazılması zordur.
- Özel donanıma ve işletim sistemine bağımlıdırlar.

#### Kullanım Alanları:

- Gömülü sistemler (mikro denetleyiciler)
- İşletim sistemi çekirdekleri
- Aygıt sürücüler
- Performansın kritik olduğu uygulamalar

### 2. Orta Seviyeli Diller

- **C Dili:** Düşük seviyeli dillerin hızına ve yüksek seviyeli dillerin okunabilirliğine sahip dengeli bir dildir.
- **C++ Dili:** C dilinin nesneye yönelik programlama özelliklerini de içeren bir genişlemesidir.

#### Özellikleri:

- Düşük ve yüksek seviyeli diller arasında bir köprü görevi görürler.
- Hem donanım kontrolü hem de yüksek seviyeli soyutlamalar sunarlar.
- Geniş kullanım alanları vardır.
- C++ gibi bazı diller, düşük seviyeli programlamaya izin verirken, aynı zamanda yüksek seviyeli özellikleri de destekler.

#### Kullanım Alanları:

- Sistem programcılığı
- Oyun geliştirme
- Bilimsel hesaplamalar
- Uygulama geliştirme

### 3. Yüksek Seviyeli Diller

- **Python:** Okunabilirliği ve basitliği ile bilinen, günümüzde en popüler dillerden biridir.

- **Java:** Geniş kütüphane desteği ve platform bağımsızlığı ile bilinir.
- **C#:** Microsoft tarafından geliştirilen, .NET Framework üzerinde çalışan nesneye yönelik bir dildir.

#### Özellikleri:

- İnsan diline daha yakın bir yapıya sahiptirler.
- Soyutlama seviyesi yüksektir, yani programcı donanımın detaylarıyla uğraşmak zorunda kalmaz.
- Genelde yorumlayıcı veya derleyici kullanırlar.
- Hızlı uygulama geliştirmeye olanak tanırırlar.

#### Kullanım Alanları:

- Web geliştirme
- Veri analizi
- Yapay zeka
- Mobil uygulama geliştirme
- Bilimsel hesaplamalar

#### Seviyeler Arasındaki Farklar

| Özellik        | Düşük Seviyeli                       | Orta Seviyeli                         | Yüksek Seviyeli              |
|----------------|--------------------------------------|---------------------------------------|------------------------------|
| Soyutlama      | Düşük                                | Orta                                  | Yüksek                       |
| Performans     | Yüksek                               | Orta                                  | Düşük                        |
| Okunabilirlik  | Düşük                                | Orta                                  | Yüksek                       |
| Kullanım Alanı | Gömülü sistemler, işletim sistemleri | Sistem programcılığı, oyun geliştirme | Web geliştirme, veri analizi |

#### Özetle:

Programlama dillerinin seviyesi, bilgisayarın donanımına ne kadar yakın olduğuyla ve programcının soyutlama düzeyiyle ilgilidir. Düşük seviyeli diller donanıma daha yakındır ve daha fazla kontrol sağlar, ancak yazılması ve anlaşılması zordur. Yüksek seviyeli diller ise daha soyuttur ve programcının işini kolaylaştırır, ancak performans açısından düşük seviyeli dillere göre dezavantajlı olabilirler.

### Hangi Dili Seçmeliyim?

Seçeceğiniz dil, projenizin gereksinimlerine ve sizin bilgi seviyenize bağlıdır. Örneğin:

- **Performans kritik uygulamalar:** C veya C++
- **Sistem programcılığı:** C veya C++
- **Web geliştirme:** Python, JavaScript, PHP
- **Mobil uygulama geliştirme:** Swift (iOS), Kotlin (Android)
- **Veri analizi:** Python, R
- **Yapay zeka:** Python

**Önemli Not:** Bu sınıflandırma genel bir çerçevedir ve bazı diller birden fazla kategoriye sığabilir. Örneğin, C++ hem orta hem de yüksek seviyeli özelliklere sahiptir.

### Daha Fazla Bilgi İçin:

Programlama dilleri hakkında daha detaylı bilgi almak için internet üzerindeki kaynakları ve eğitimleri inceleyebilirsiniz.

**Bu konuda başka sorularınız var mı?**

## Programlama Dilinin Okunabilirliği Nedir?

**Programlama dilinin okunabilirliği**, bir yazılım kodunun ne kadar kolay anlaşılır olduğunu ifade eder. Yani, bir başka programcı veya hatta aynı kodu daha sonra tekrar inceleyecek olan kişinin, kodun mantığını ve işleyişini minimum çaba ile kavrayabilmesi anlamına gelir.

### Neden Önemlidir?

- **Bakım ve Güncelleme:** Yazılımlar sürekli gelişir ve değişir. Okunabilir bir kod, bu değişiklikleri yapmak için gereken süreyi ve çabayı önemli ölçüde azaltır.
- **İşbirliği:** Birden fazla kişinin çalıştığı projelerde, okunabilir kod, ekip üyelerinin birbirlerinin kodlarını anlamalarını kolaylaştırır ve böylece daha verimli bir çalışma ortamı sağlar.
- **Hata Ayıklama:** Hataları bulmak ve düzeltmek için kodun okunabilir olması şarttır. Okunabilir bir kodda, hatalar daha hızlı tespit edilir ve çözümüendirilir.
- **Yeni Başlayanlar İçin Öğrenme:** Okunabilir bir kod, programlamaya yeni başlayanların konuyu daha kolay anlamalarına yardımcı olur.

### Okunabilirliği Etkileyen Faktörler:

- **Söz Dizimi:** Dilin söz dizimi ne kadar basit ve tutarlıysa, kod o kadar okunabilir olur.
- **İsimlendirme:** Değişken, fonksiyon ve sınıf gibi öğelere verilen isimler anlamlı ve açıklayıcı olmalıdır.
- **Gömülü Yorumlar:** Kodun kritik bölümlerine eklenen yorumlar, kodun ne yaptığını anlamayı kolaylaştırır.
- **Girinti ve Biçimlendirme:** Tutarlı bir girinti ve biçimlendirme, kodun görsel olarak daha düzenli olmasını sağlar ve okunabilirliği artırır.
- **Kütüphane ve Framework Kullanımı:** İyi tasarlanmış kütüphane ve framework'ler, kodun daha az karmaşık olmasını sağlar.
- **Kodun Yapısı:** Modülerlik, fonksiyon ayrımı gibi yazılım tasarım prensipleri, kodun okunabilirliğini artırır.

### Örnek:

Python

```
# Okunabilir kod örneği
def faktoriyel(sayi):
    """Bir sayının faktöriyelini hesaplar."""
    if sayi == 0:
        return 1
    else:
        return sayi * faktoriyel(sayi - 1)
```

Python

```
# Okunabilirliği düşük kod örneği
def f(x):
    if x==0: return 1
    else: return x*f(x-1)
```

### Sonuç:

Programlama dilinin okunabilirliği, yazılım geliştirme sürecinin her aşamasında önemli bir rol oynar. Okunabilir bir kod yazmak, hem geliştiricilerin verimliliğini artırır hem de yazılımın uzun ömürlü olmasını sağlar. Bu nedenle, programcılar kod yazarken okunabilirliğe özen göstermelidirler.

### Ek Bilgiler:

- **Okunabilirliği Artırmaya Yönelik Araçlar:** Bazı araçlar, kodun karmaşıklığını ölçerek ve iyileştirme önerileri sunarak okunabilirliği artırmaya yardımcı olabilir.
- **Kod İncelemesi:** Kod incelemesi, birden fazla kişinin kodunu gözden geçirerek okunabilirliği ve kalitesini artırmak için kullanılan bir yöntemdir.
- **Stil Kılavuzları:** Birçok programlama dili için belirlenmiş stil kılavuzları, kod yazımında tutarlılık sağlamak ve okunabilirliği artırmak için kullanılır.

**Başka sorularınız var mı?** Örneğin, belirli bir programlama dilinin okunabilirliği hakkında daha fazla bilgi almak isteyebilirsiniz.