

Programlama Dili

Giriş

Bilgisayar sistemlerinin anlaşılabilir şekilde yönlendirilebilmesi, insan ile makine arasındaki iletişimin tanımlı kurallar ve yapı taşlarıyla gerçekleştirilmesini gerektirir. Bu iletişim, programlama dilleri aracılığıyla sağlanır. Programlama, yalnızca bir metin yazma eylemi değil; aynı zamanda algoritmik düşünme, veri yönetimi ve sistemli problem çözme sürecidir. Yazılım geliştirme süreçlerinin temelini oluşturan bu yapıların iyi anlaşılması, çağın gereksinimlerine uygun nitelikli sistemler üretmenin ilk adımıdır.

Programlama Dili Nedir?

Programlama dili; bilgisayarlara, elektronik sistemlere veya diğer dijital aygıtlara ne yapmaları gerektiğini anlatan yapay dillerdir. Bu dillerin amacı, insanların ifade ettiği mantıksal düşüncüyü makinenin anlayabileceği komutlara çevirmektir. Programlama dilleri; değişkenler, veri tipleri, kontrol yapıları, döngüler, fonksiyonlar ve nesneler gibi bileşenlerle birlikte belirli sözdizimi kurallarıyla yapılandırılmıştır. Her bir programlama dili, farklı ihtiyaçlara cevap vermek üzere tasarlanmıştır.

Programlama Dilleri Nelerdir?

Farklı alanlara hizmet eden çok sayıda programlama dili mevcuttur. Bu dillerin bazıları genel amaçlı, bazıları ise daha özel görevler için optimize edilmiştir:

- **Python:** Basit sözdizimiyle yeni başlayanlar için idealdir. Veri bilimi, makine öğrenimi, yapay zekâ, web geliştirme ve otomasyon gibi birçok alanda kullanılır.
- **Java:** Platformdan bağımsızlığı sayesinde kurumsal yazılımlarda ve Android geliştirmede yaygın tercih edilir.
- **C:** Sistem düzeyinde yazılım geliştirmek, donanıma yakın işlem yapmak için kullanılır. İşletim sistemlerinin çekirdekleri genellikle C ile yazılır.
- **C++:** C dilinin nesne yönelimli özelliklerle genişletilmiş hâlidir. Oyun motorları, grafik işlem uygulamaları gibi yüksek performanslı işler için uygundur.
- **C#:** Microsoft teknolojileriyle çalışan masaüstü ve oyun uygulamalarında (örneğin Unity) yaygındır.

- **JavaScript:** Web tarayıcılarında çalışan dinamik içerikleri oluşturan temel dildir. Front-end (ön yüz) geliştirme başta olmak üzere web uygulamalarının vazgeçilmezidir.
- **PHP:** Web geliştirme amaçlı sunucu taraflı çalışan bir dildir.
- **Kotlin:** Android uygulama geliştirmede Java'nın modern alternatifi olarak geliştirilmiştir.
- **Swift:** Apple ekosisteminde (iOS, macOS) uygulama geliştirmek için kullanılan güçlü bir dildir.

Programlama Dillerinin Makineye ve İnsana Yakınlık Sıralaması

Programlama dilleri soyutlama seviyelerine göre dört temel sınıfa ayrılır:

1. **Yüksek Seviye Diller (İnsana Yakın):** Sözdizimi insan diline yakın olup okunması ve yazılması kolaydır.
 - a. Örnekler: Python, Java, Ruby, C#
2. **Orta Seviye Diller:** Hem düşük seviye hem de yüksek seviye özellikleri barındırır. Donanımla daha fazla etkileşim sağlar.
 - a. Örnekler: C, C++
3. **Düşük Seviye Diller (Makineye Yakın):** Donanım üzerinde doğrudan kontrol sağlar; öğrenilmesi ve yazılması zordur.
 - a. Örnek: Assembly
4. **Makine Dili:** Bilgisayarın doğrudan anlayabildiği 0 ve 1'lerden oluşan dil.

Bu sıralama, dilin donanım üzerindeki doğrudan kontrol yeteneği ile insan okuyucusu tarafından anlaşılabilirliği arasındaki ilişkiye dayanır.

Derleyici (Compiler) ve Yorumlayıcı (Interpreter) Nedir?

Programlama dillerinde yazılan kodlar, doğrudan makine tarafından anlaşılamaz. Bu nedenle kodların, makine diline çevrilmesi gerekir. Bu çeviri işlemi, derleyiciler veya yorumlayıcılar tarafından yapılır:

- **Derleyici (Compiler):** Kaynak kodu bir defada alır ve tamamını makine diline çevirerek çalıştırılabilir bir dosya oluşturur. Derleme aşamasında oluşabilecek hatalar önceden tespit edilir.
 - Örnek: C, C++, Java (javac)
- **Yorumlayıcı (Interpreter):** Kodları satır satır okuyarak çalıştırır. Hatalar ancak o satıra gelindiğinde fark edilir.

- Örnek: Python, JavaScript

Bazı diller (örneğin Java), her iki yöntemi birlikte kullanabilir: önce derlenip sonra yorumlanarak çalıştırılabilir.

IDE (Entegre Geliştirme Ortamı) Nedir?

IDE (Integrated Development Environment), programcıların yazılım geliştirme sürecinde ihtiyaç duyduğu çeşitli araçları bir arada sunan yazılımlardır. IDE'ler; kod düzenleyici, derleyici/yorumlayıcı, hata ayıklayıcı (debugger), sürüm kontrol sistemleri ve proje yönetim araçlarını entegre şekilde sağlar.

Yaygın IDE Örnekleri:

- **Visual Studio:** C#, C++, .NET platformları için güçlü bir ortam sağlar.
- **PyCharm:** Python için özel olarak geliştirilmiştir.
- **IntelliJ IDEA:** Java ve Kotlin için gelişmiş destek sunar.
- **Visual Studio Code:** Hafif, modüler bir IDE'dir. Birçok dili destekler.
- **Xcode:** iOS ve macOS uygulama geliştirme için Apple tarafından geliştirilmiştir.

IDE'ler, yazılım geliştirme sürecini hızlandırır, hata riskini azaltır ve geliştiricinin üretkenliğini artırır.

Parametre Nedir?

Parametreler, fonksiyon ya da metotlara gönderilen girdilerdir. Bir işlemin, dışarıdan verilen değere göre farklı şekilde davranmasını sağlar. Parametreler, yazılıma esneklik ve yeniden kullanılabilirlik kazandırır. Örneğin:

```
def kare_al(x):  
    return x * x
```

Burada x bir parametredir ve fonksiyon bu parametreye göre çalışır.

Bit Nedir?

Bit, bilgisayarın anlayabileceği en küçük veri birimidir. "Binary digit" kelimesinden türetilmiştir ve sadece iki değer alabilir: **0** ya da **1**.

- 8 bit = 1 byte

- 1024 byte = 1 kilobyte (KB)

Bilgisayar sistemlerinde tüm veriler, dosyalar ve komutlar bit'lerin bir araya gelmesiyle temsil edilir.

0-1 Parametresi Nedir?

Sayısal sistemlerde 0 ve 1 değerleri genellikle ikili mantıkta iki ayrı durumu ifade eder:

- **0:** Kapalı / Pasif durum
- **1:** Açık / Aktif durum

Bu mantıkla çalışan birçok elektronik sistem vardır. Örneğin, bir uzaktan kumanda ile televizyonun sesi açılmak istendiğinde gönderilen sinyalin değeri 1 olabilir; bu sinyal, alıcı tarafından ses açma komutu olarak algılanır. 0 ise hiçbir işlem yapılmaması gerektiğini ifade edebilir.

Algoritma Nedir? (Yazılım ve Programlama Bağlamında)

Algoritma, belirli bir problemi çözmek veya belirli bir görevi yerine getirmek için tanımlanmış sonlu adımlar bütünüdür. Yazılım geliştirme sürecinde, algoritmalar problemi çözmek için düşünsel çerçeveyi oluşturur; bu çerçeve daha sonra programlama dili ile kodlanır.

Günlük Yaşamdan Örnek – Çay Koyma:

1. Suyu kaynat.
2. Demliğe çay koy.
3. Kaynar suyu demliğe dök.
4. 10 dakika bekle.
5. Çayı bardağa koy.

Programlama Örneği – Python:

```
def topla(a, b):  
    return a + b
```

Bu örnek, toplama işlemi için oluşturulmuş basit bir algoritmanın programlama diliyle ifadesidir.

Veri Tabanı Nedir?

Veritabanı, dijital verilerin yapılandırılmış bir şekilde saklandığı sistemdir. Tablolar, sütunlar, satırlar şeklinde organize edilmiş veriler üzerinde arama, güncelleme, silme gibi işlemler yapılabilir. Kullanıcı bilgileri, ürün katalogları, sipariş geçmişi gibi dinamik veriler veritabanlarında saklanır.

Veritabanları; veriye hızlı erişim, güvenli saklama ve eş zamanlı çoklu kullanıcı desteği gibi avantajlar sağlar.

Programlama Dillerinde Veritabanının Önemi

Modern yazılımların büyük bir kısmı, veri ile çalışır. Bu verinin kalıcı olarak saklanması ve gerektiğinde hızlıca erişilebilmesi için veritabanları kullanılır. Programlama dilleri, veritabanlarına bağlanarak aşağıdaki işlemleri gerçekleştirir:

- Veri ekleme (INSERT)
- Veri güncelleme (UPDATE)
- Veri silme (DELETE)
- Veri çekme (SELECT)

Örneğin, bir e-ticaret sitesinin ürünleri PHP ile yazılmış bir arayüzden sorgulanıp MySQL veritabanından çekilebilir. Bu ilişki olmadan, kullanıcıya dinamik bilgi sunmak mümkün değildir.

Veri Çekmek Nedir?

Veri çekmek; veritabanında kayıtlı olan bilgilerin program aracılığıyla sorgulanıp kullanıcıya sunulması işlemidir. Genellikle SQL (Structured Query Language) kullanılarak gerçekleştirilir.

Örnek Sorgu:

```
SELECT * FROM ogrenciler WHERE yas > 18;
```

Bu sorgu, yaş değeri 18'den büyük olan tüm öğrencilerin bilgilerini çeker. Bu veriler programlama dili üzerinden kullanıcıya sunulur veya işlenir.

Sonuç

Programlama; algoritmik düşünme, veri yönetimi ve sistem geliştirme becerilerinin bütünüdür. Programlama dilleri, veritabanları, derleyiciler, algoritmalar ve IDE'ler gibi bileşenlerin her biri, yazılım üretiminin farklı bir boyutunu oluşturur. Bu kavramların birlikte ve doğru şekilde kullanılması, modern teknolojilerin temel yapı taşlarını oluşturur. Yazılım geliştirmenin temelini anlamak, dijital dünyada etkili ve üretken olmanın ön koşuludur.

Bu metin, BTK Akademi "Bilgi Teknolojilerine Giriş" eğitiminden edinilen bilgilerle **Arda Karadağ** tarafından hazırlanmıştır.