



Bölüm 1: Bilgisayarlar

JAVA ile Nesne Yönelimli Programlama



Temel Parçalar

- Bir bilgisayar, temelde birkaç önemli parçadan oluşur.
 - **İşlemci (CPU):** Bilgisayarın beyni olarak nitelendirilebilir. Tüm hesaplamaları yapmakla sorumludur.
 - **Bellek (RAM):** Geçici verileri saklar, dolayısıyla kapasite ve hız gibi özellikleri bilgisayarın çalışma hızını etkiler.
 - **Sabit Disk:** Kalıcı verileri saklar. Dosya, uygulama gibi verilerin depolanmasından sorumludur.



Bilgisayarın İşleyişi

- Temel adımlar:
 - İşlemciye talimatlar gönderilir.
 - İşlemci, talimatları sırayla işler.
 - İşlem sonuçları kullanıcıya veya diğer yazılımlara gösterilir.
- Programlama, işlemcinin anlayabileceği bir dilde talimatlar yazma sürecidir.
 - Bilgisayarların belirli görevleri yerine getirmeleri için kullanılır.



Bir Bilgisayar Nasıl İnşa Edilir?

- **Sayılar (Numbers)**
- Harf ve Dizgeler (Letters and Strings)
- Yapılandırılmış Bilgi (Structured Information)
- Saklanan Program Kavramı (Stored Program Concept)
- von Neumann Mimarisi (von Neumann Architecture)



Sayılar

- Bir bilgisayarın temel işlevi elektrik sinyallerini belirli kurallara göre manipüle etmektir. Bu kurallar sayesinde bilgisayarlar birçok işlevi gerçekleştirir.
- Elektrik sinyallerini kullanıcılar olarak bizlerin anlayabileceği sayılar ve sembollerle ilişkilendirmek önemlidir. Bu sayede bilgisayarlar, bize anlamlı bilgiler sunabilir.
- Elektrik sinyalleri, ikili (binary) temsili kullanarak bilgisayarlar aracılığıyla bilgileri işler.



Sayılar

- Bilgisayarlar, tam sayıları temsil etmek için 2'nin üssü olan sayıların kombinasyonlarını kullanırlar. Bu sistem "ikili" veya "taban 2" olarak adlandırılır.
- Her bir basamak (bit), 0 veya 1'i temsil eder. Örneğin:
 - 0 veya 1
 - Yanlış veya Doğru
 - Kapalı veya Açık
 - Düşük voltaj veya Yüksek voltaj
- Bu ikili sistem, bilgisayarlar için çok temel bir yapı taşıdır ve hemen hemen tüm işlemler bu temel üzerine inşa edilir.



Sayılar

- 4 ardışık üssü kullanılarak (2^0 , 2^1 , 2^2 , 2^3), 0 ile 15 arasındaki tüm tam sayılar temsil edilebilir. Bu, her bir üssü 0 veya 1 kez kullanarak yapılır. Örneğin:
 - $13_{10} = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1101_2$; başka bir deyişle, 1101 ikili tabanda 13 ondalık sayısına denk gelir.
- Sayıları temsil etmek için farklı sayı sistemleri kullanılabilir. Örneğin:
 - 603 ondalık tabanda, $603_{10} = 6 \cdot 10^2 + 0 \cdot 10^1 + 3 \cdot 10^0$ olur.
 - 207 sekizlik tabanda, $207_8 = 2 \cdot 8^2 + 0 \cdot 8^1 + 7 \cdot 8^0$ olur.



Sayılar

- Herhangi bir taban ($b \geq 2$) seçildiğinde,
 - 0 ile $b^d - 1$ arasındaki her pozitif tam sayı,
 - d adet basamak kullanarak benzersiz bir şekilde temsil edilebilir.
 - Bu basamakların katsayıları 0'dan $b-1$ 'e kadar olan değerleri alabilir.
- Modern bir 64-bit bilgisayar, $2^{64} - 1$ 'e kadar olan tam sayıları temsil edebilir.
- Bu, büyük veri işlemleri ve hesaplamalar için çok önemlidir.



Sayılar

- Herhangi bir sayı sistemi, aritmetik işlemleri gerçekleştirirken benzer temel kuralları takip eder.
- Toplama işlemi aynı temel ilkelere dayanır: basamak değerleri, taşıma işlemleri ve sonucun doğru şekilde hesaplanması. Hangi sayı tabanı kullanılırsa kullanılsın, bu ilkeler geçerlidir.
- Toplama İşlemi: 10 Tabanı (Ondalık):
 - $57 + 28 = 85$
- Toplama İşlemi: 2 Tabanı (İkili):
 - $00111001 + 00011100 = 01010101$



Sayılar

- Negatif tam sayıları temsil etmek için bilgisayarlar genellikle «ikiye tümler» (two's complement) adı verilen bir sistem kullanır.
- Bu sistem, pozitif bir sayının bitlerini tersine çevirip sonra 1 eklemeyi içerir.
- Sonuç, negatif sayının temsili olur.
- Örneğin;
 - Elimizde 8-bitlik bir bilgisayar var ve 3 sayısını temsil etmek istiyoruz. 3, ikili tabanda şu şekilde gösterilir: 00000011.
 - Negatif 3 sayısını temsil etmek için, pozitif 3 sayısının bitlerini tersine çeviririz. Yani 00000011'den 11111100 yaparız.
 - Ardından 1 ekleriz, yani $11111100 + 1 = 11111101$.



Sayılar

- Ondalık tabanda sınırlı sayıda (örneğin 8) rakam (0-9) kullanılarak sınırlı sayı temsil edilebilir. Ancak bazen çok büyük veya çok küçük sayıları temsil etmemiz gerekebilir. Bu noktada kesirli ve üssel bölüm devreye girer.
- Kesirli kısımları ve üsleri ayrı olarak ele alınarak, büyük veya küçük sayılar daha etkili bir şekilde temsil edilebilir.
- Örneğin, 31415901 sayısını ele alalım.
 - İlk altı rakam (314159) kesirli kısmı temsil eder ve noktadan sonraki rakamlar kesirli kısmı ifade eder.
 - Son iki rakam (01), sayının üssünü temsil eder. 01, 10'un kuvveti olan 10^1 'i ifade eder. Yani, sonuç $0.314159 \times 10^1 = 3.14159$ olur.



Bir Bilgisayar Nasıl İnşa Edilir?

- Sayılar (Numbers)
- **Harf ve Dizgeler (Letters and Strings)**
- Yapılandırılmış Bilgi (Structured Information)
- Saklanan Program Kavramı (Stored Program Concept)
- von Neumann Mimarisi (von Neumann Architecture)



Harf ve Dizgeler

- Harfleri sayısal olarak temsil etmek, bilgisayarlar için bir gerekliliktir. Bu, metin tabanlı verileri işlememize ve iletişim kurmamıza yardımcı olur.
- ASCII, Amerikan Ulusal Standartlar Enstitüsü (ANSI) tarafından belirlenen bir kodlama standardıdır.
- ASCII, büyük ve küçük harfleri, rakamları ve belirli özel karakterleri sayısal olarak temsil etmek için kullanılır.
 - 8-bit bir kod olduğundan toplamda 256 farklı sembolü temsil edebilir.
- ASCII'nin en büyük sınırlamalarından biri, sadece 256 farklı sembolü temsil edebilmesidir. Çok sayıda farklı dilde kullanılan karakterlerin temsili için yetersizdir. Başka dillerdeki harfleri, sembolleri veya işaretleri temsil etmek için farklı kodlamalar ve standartlar kullanmak gerekebilir.



Harf ve Dizgeler

- Unicode, Uluslararası Standartlar Organizasyonu (ISO) tarafından oluşturulan bir sistemdir.
- Unicode, her dildeki harf, sembol, ve işareti temsil edebilmek için 16-bit kodlama kullanır. Bu, çok sayıda dilin ve karakterin temsil edilmesini mümkün kılar.
- Özellikle İngilizce metinlerde Unicode bazen gereğinden fazla bellek alanı kullanabilir. Bu nedenle, ISO, "Unicode Dönüşüm Formatları" (UTF) adı verilen alternatif kodlama yöntemi tanımlamıştır.
- UTF-8, Unicode karakterlerini temsil etmek için dinamik bir yaklaşım kullanır. Sık kullanılan karakterler daha az bellek kullanırken, nadir kullanılan karakterler daha fazla bellek kullanır.



Harf ve Dizgeler

- Emojiler, iletişimimizi zenginleştiren ve duyguları, ifadeleri veya nesneleri hızla aktarmamıza yardımcı olan küçük simgelerdir.
- Emojiler, bilgisayarlar tarafından metin olarak işlenir ve karakterler gibi temsil edilir. Ancak, emojiler çok daha karmaşık ve renklidirler, bu yüzden onları temsil etmek için özel bir standart geliştirilmiştir.
- Emojilerin temsil edilmesi için Unicode Consortium adlı organizasyon tarafından standartlar belirlenmiştir.
- Bu standartlar, her bir emojiyi benzersiz bir kodla temsil etmek için kullanılır. Böylece, emojiler her cihazda veya uygulamada aynı şekilde görünür.



Harf ve Dizgeler

- Dizge (string), metin veya karakterlerin bir sırasını temsil eden bir veri türüdür. Örneğin, "Merhaba Dünya!" bir dizgedir.
- Dizgeler, metin işleme ve iletişimde sıkça kullanılır.
- Dizgeler, sayılarla temsil edilir.
- Dizge içindeki her karakterin (harf, rakam, sembol) bir sayı değeri vardır.
- Ayrıca, dizgenin uzunluğunu belirlemek için bir "uzunluk alanı" (length field) kullanılır.
- Örneğin, «çikolata» dizgesini ele alalım. ASCII kodlamasına göre, bu dizge şu sayı dizisine çevrilebilir: 231, 105, 107, 111, 108, 97, 116, 97.



Bir Bilgisayar Nasıl İnşa Edilir?

- Sayılar (Numbers)
- Harf ve Dizgeler (Letters and Strings)
- **Yapılandırılmış Bilgi (Structured Information)**
- Saklanan Program Kavramı (Stored Program Concept)
- von Neumann Mimarisi (von Neumann Architecture)



Yapılandırılmış Bilgi

- Sayı dizileri, bilgiyi anlamamıza ve bilgisayarlarla etkileşimde bulunmamıza yardımcı olan güçlü bir araçtır.
- Her türlü bilgiyi sayılarla temsil edebilmek, bilgi işlem dünyasının temel taşlarından biridir.
 - **Resimler:** Her piksel, kırmızı, yeşil ve mavi bileşenlerin miktarını temsil eden üç sayıdan oluşur. Örneğin, bir mavi piksel için (0, 0, 255) gibi.
 - **Sesler:** Ses, havadaki "ses basınç seviyeleri" olarak temsil edilir. Bu, bir sese ait titreşimlerin zaman içindeki kaydını ifade eder.
 - **Filmler:** Bir film, saniyede 24 veya 30 kare gibi hızlarda ardışık resimlerin dizisiyle temsil edilir. Her kare, resimlere benzer şekilde sayı dizileriyle temsil edilir.



Bir Bilgisayar Nasıl İnşa Edilir?

- Sayılar (Numbers)
- Harf ve Dizgeler (Letters and Strings)
- Yapılandırılmış Bilgi (Structured Information)
- **Saklanan Program Kavramı (Stored Program Concept)**
- von Neumann Mimarisi (von Neumann Architecture)



Saklanan Program Kavramı

- Modern bilgisayarların temel ilkesini oluşturur. Bu ilkeye göre, bilgisayarın belleğinde saklanan komutlar ve veriler programları oluşturur.
- ENIAC'tan farklı olarak, EDVAC'da komutlar ve veriler bellekte sıralı bir şekilde saklanır ve bu sayede programlar daha esnek ve genişletilebilir hale gelmiştir.
- Bilgisayar belleğinde saklanan komutlar ve veriler, programın işleyişini belirler. Her komut, bir görevi veya işlemi temsil eder. Bu komutlar ardışık olarak işlenir.
- Komutlar ardışık olarak işlense de, bazen bir koşullu komut, bir koşul sağlandığında programın sıradan farklı bir komuta atlamasına neden olabilir. Bu, programların belirli şartlara göre farklı yollar izlemesini sağlar.



Saklanan Program Kavramı

- Bellek, bilgisayarın verileri ve programları geçici olarak sakladığı yerdir.
- Bellek hiyerarşisi genellikle üç seviyede incelenir:
 - **İç (L1) Cache:** En hızlı ve en yakın önbellek. İşlemcinin hemen erişebildiği yerdir.
 - **Orta (L2) Cache:** L1 önbelleği destekler ve daha büyük bir kapasiteye sahiptir.
 - **Ana Bellek (RAM):** Daha büyük bir depolama alanı sağlar, ancak L1 ve L2 önbelleklerine göre daha yavaş çalışır.
- Bellek hiyerarşisi, bilgisayarın performansını büyük ölçüde etkiler. Daha hızlı ve işlemciye yakın bellekler, daha hızlı veri sağlar, bu da işlemlerin hızlanmasına neden olur.



Saklanan Program Kavramı

- «Al-Çöz-Çalıştır» (Fetch-Decode-Execute) döngüsü, bilgisayarların işlemci tarafından programların nasıl yürütüldüğünü tanımlar.
- Bu döngü üç adımdan oluşur:
 - Al,
 - Çöz ve
 - Çalıştır.
- "Al-Çöz-Çalıştır" döngüsü sürekli olarak tekrarlanır.
- İşlemci sıradaki komutu alır, çözer ve çalıştırır.
- Ardından bir sonraki komutu alır ve bu döngü devam eder.



Saklanan Program Kavramı

- **Adım 1: Al (Fetch):**
 - İlk adımda, işlemci programın bir sonraki komutunu bellekten alır.
- **Adım 2: Çöz (Decode):**
 - İkinci adımda, alınan komut çözülür. Her komutun belirli bir işlevi vardır ve işlemci bu işlemin ne olduğunu anlar.
- **Adım 3: Çalıştır (Execute):**
 - Üçüncü adımda, çözülen komut çalıştırılır. İşlemci komutun gerektirdiği işlemi gerçekleştirir. Bu işlem, verileri işlemek, sonuçları hesaplamak veya başka bir işlemi başlatmak olabilir.



Saklanan Program Kavramı

- Merkezi İşlem Ünitesi veya kısaca CPU, bilgisayarın "beyni" olarak düşünülebilir. Burada, tüm hesaplamalar ve işlemler gerçekleşir.
- CPU, iki temel bileşenden oluşur:
 - Aritmetik işlem birimi (ALU) ve
 - Kontrol Birimi.
- Aritmetik işlem birimi, hesaplamaları gerçekleştiren parçadır, Kontrol Birimi ise komutları yönlendirir ve işlemleri düzenler.
- CPU, toplama, çarpma gibi aritmetik işlemleri gerçekleştirmek için parçalara sahiptir. Aynı zamanda, düşük kapasiteli «yazmaç» adı verilen alanlara sahiptir. Bu yazmaçlar, geçici verileri saklamak için kullanılır.



Saklanan Program Kavramı

- Veri yolu veya kısaca BUS, birçok paralel telleri içeren bir iletişim yolu sistemidir.
- Veri yolu, üç ana türde sinyali taşır:
 - **Adres Sinyalleri:** Belirli bir bellek veya cihazı hedeflemek için kullanılır.
 - **Veri Sinyalleri:** Gerçek verileri taşır, bu veriler işlenir veya depolanır.
 - **Kontrol Sinyalleri:** Veri yolu üzerindeki işlemi yönlendiren sinyallerdir. Örneğin, yazma veya okuma işlemlerini kontrol eder.



Saklanan Program Kavramı

- Bellek, bilgisayarın içinde verilerin ve talimatların saklandığı depolama alanıdır.
- Bir program, uzun bir talimat listesini içerir. Bu talimatlar, bilgisayarın ne yapması gerektiğini belirler.
- Programlar bellekte saklanır ve sırasıyla işlenir.
- Bir programdaki talimatlar, bellekten tek tek alınır ve işlemci içindeki özel bir alana «yazmaçlara» yüklenir. İşlemci bu yazmaçlardaki talimatları sırayla çalıştırır.
- Bu yazmaçlar, aritmetik ve mantıksal işlemler için kullanılır.
- En önemli yazmaçlardan biri, **program sayacıdır**. İşlemciye sırada hangi talimatın olduğunu ve hangi talimatın işlenmesi gerektiğini söyler.



Saklanan Program Kavramı

- Talimatlar, bellekte sayılarla kodlanır. Her talimat, bir bellek adresine ve bu adresteki içeriğe sahiptir. İşlemci, bu bellek adreslerini okur ve bu adresteki talimatları çalıştırır.
- Her bellek adresi, bellek hücresine işaret eder. Burada, bir talimatın veya verinin kodlanmış şekli bulunur. Örneğin, 200 bellek adresi "ADD to R1" talimatını içerir.
- Bellek hem verileri hem de program talimatlarını saklar. Veriler, işlemci tarafından işlenirken, program talimatları işlemci tarafından sırayla çalıştırılır.
- Örneğin, 200 bellek adresinde "ADD to R1" talimatı, 201 bellek adresinde "102" gibi bir veri değeri olabilir. İşlemci bu talimatları adres sırasına göre çalıştırır.
- Bazı talimatlar işlemcinin belirli bir sırayı atlamasını sağlar. Örneğin, 204 bellek adresindeki "JUMP 7 bytes" komutu, işlemcinin 7 bayt sonrasındaki talimatlara atlamasını sağlar.



Bir Bilgisayar Nasıl İnşa Edilir?

- Sayılar (Numbers)
- Harf ve Dizgeler (Letters and Strings)
- Yapılandırılmış Bilgi (Structured Information)
- Saklanan Program Kavramı (Stored Program Concept)
- **von Neumann Mimarisi (von Neumann Architecture)**



von Neumann Mimarisi

- 8-bit bir bilgisayarımız olduğunu varsayalım. Bu bilgisayar, sadece dört temel komutla çalışır: toplama, çıkarma, çarpma ve bölme.
- Her komut, bir işlem kodu veya opcode ile temsil edilir. İşlem kodları, hangi işlemin yapılacağını belirtir. Örneğin, toplama komutu için bir işlem kodu, çarpma için başka bir işlem kodu kullanılır.
- Bilgisayarımızın 4 yazmacı ve 256 adet 8-bit bellek hücresi vardır.
- Her komut, belirli bir şekilde kodlanır. Bir komutun kodlaması şu şekilde olabilir: İlk iki bit işlem kodunu temsil eder, sonraki iki bit hedef yazmacı (destination register) ve sonraki dört bit ise iki işlenen yazmacı temsil eder.
- Örneğin, "3 0 2" anlamına gelen "register 2 ve register 0 içeriğini topla ve sonucu register 3'e kaydet" komutu, 00110010 şeklinde kodlanabilir.



von Neumann Mimarisi

- Programlama, farklı soyutlama seviyelerinde gerçekleşebilir.
 - **Yüksek Seviye Dil (High-level language):** Problem alanına daha yakın bir soyutlama seviyesine sahiptir. Programcılara daha fazla üretkenlik ve taşınabilirlik sağlar. İnsanlar için daha anlaşılabilir ve okunabilir yazılımlar oluşturmaya yardımcı olur.
 - **Çevirici Dili (Assembly language):** Bilgisayar programlarını metin temelli komutlarla ifade eden bir düzeydir. Daha sonra donanım temsili olarak çevrilmek üzere kullanılır. Daha düşük seviyeli bir soyutlama sunar.
 - **Donanım Temsili (Hardware Representation):** Bilgiler sadece ikili sayılar (bitler) ile temsil edilir. Komutlar ve veriler, doğrudan donanım tarafından anlaşılır.



von Neumann Mimarisi

- Çevirici dili, belirli bir bilgisayar mimarisi için özelleştirilmiştir. Yani, bir bilgisayarın çevirici dili, başka bir bilgisayarın dilinden farklı olabilir.
- Birkaç örnek komut:
 - `MOV AL, 61h`: AL yazmacına 61h değerini taşı
 - `MOV AX, BX`: AX yazmacına BX yazmacındaki değeri taşı
 - `ADD EAX, 10`: EAX yazmacına 10 değerini ekle
 - `XOR EAX, EAX`: EAX yazmacını sıfırla
- Çevirici dili, bilgisayarın alt seviyede nasıl çalıştığını anlamak için önemlidir.
- Performans optimizasyonu ve donanım ile etkileşim için kullanılır.



Soru

Hangi kavram, bilgisayarda sayıları temsil etmek ve işlemek için kullanılan 0 ve 1'lerin bir kombinasyonunu içerir?

- a) ASCII Kodlaması
- b) İkilik Temsil
- c) Unicode
- d) Montaj Dili



Soru

Bilgisayarlar, metinler ve karakterlerin temsil edilmesi için hangi kavramı kullanır?

- a) İkilik Temsil
- b) ASCII Kodlaması
- c) Hafıza Hiyerarşisi
- d) Von Neumann Mimarisi



Soru

Von Neumann mimarisi, aşağıdakilerden hangisiyle en iyi tanımlanır?

- A) Bir bilgisayarın dört ana bileşenden oluştuğu bir model: işlem birimi, bellek, giriş birimleri ve çıkış birimleri
- B) Bir bilgisayarın programları bellekte saklayabilme ve çalıştırabilme yeteneği
- C) Bir programın bilgisayarın sabit diskine kaydedilmesi ve gerektiğinde yüklenmesi
- D) Tümünün cevapları



Soru

Bir bilgisayarın işlem yapabilmesi için saklanmış program konsepti gerektirir. Bu kavramın temel özelliği nedir?

- a) İşlemcinin çok yüksek hızda çalışması
- b) Programların sabit bellekte saklanması
- c) Programların kullanıcı tarafından yazılması
- d) Programların bellekte saklanması ve çalıştırılması



Soru

ASCII kodlamasına göre, büyük harf "A" hangi sayıya karşılık gelir?

- a) 64
- b) 65
- c) 66
- d) 67



Soru

Von Neumann Mimarisi'nde, bilgisayarın programları ve verileri aynı bellek içinde saklar. Ancak bu bellek aşağıdakilerden hangisiyle sınırlıdır?

- a) Sınırsız
- b) Yüksek hızlı
- c) Sabit boyutlu
- d) Taşınabilir



Soru

Dizgeler (string), hangi tür bilgileri temsil etmek için kullanılır?

- a) Sayıları
- b) Karakterleri
- c) Kesirli sayıları
- d) Tarihleri



Soru

Bilgisayarlar, sayıları ve harf/dizgi karakterlerini saklamak için hangi temsil sistemini kullanır?

- a) Onlu sistem
- b) İkilik sistem
- c) Sekizlik sistem
- d) Onaltılı sistem



Soru

İşlemcilerde her komutun ne iş yapması gerektiği belirleyen bileşen hangisidir?

- a) ALU (Aritmetik Mantık Birimi)
- b) Yazmaçlar
- c) Giriş/Çıkış Birimi
- d) İşlemci (CPU)



SON