Bölüm 1: Giriş

JAVA ile Nesne Yönelimli Programlama

Programlama Nedir?

- Programlama, bilgisayarlarınıza talimatlar vermenin bir yoludur.
- Bilgisayarlar karmaşık görevleri hızlı bir şekilde gerçekleştirmek için programlama kullanır.
- Her sektörde programlama kullanılıyor: oyun geliştirme, sağlık sektörü, iş analitiği ve daha fazlası.
- Kendi projelerinizi oluşturabilir ve problem çözebilirsiniz.
- Problem çözme yeteneği: Programlama, zihinsel yetenekleri geliştirir.
- Programlamaya başlamak, yeni bir dünyanın kapılarını açabilir.
- Programlama dilleri, bilgisayarlarla iletişim kurmanın araçlarıdır.
- Her zaman soru sormaktan ve öğrenmeye istekli olmaktan korkmayın.

Bilgisayarlar Artık Her Yerde!

- Evimizden iş yerimize, eğlenceden eğitime kadar her yerde kullanılıyorlar.
- Akıllı telefonlar, tabletler, dizüstü bilgisayarlar, ev aletleri, arabalar ve daha fazlası.
- Bilgisayarlar hayatımızın vazgeçilmez bir parçası haline geldi.
- Programlama, bu dijital çağın dili ve anahtarıdır.
- İnternet ve dijital teknolojiler sayesinde dünya bir tıklama uzağımızda.
- Alışveriş, haberler, iletişim, eğitim - hepsi dijital dünyada.

Programlamanın Rolü

- Programlama, bilgisayarları çalıştırmak ve istediğimiz görevleri yerine getirmek için kullanılan bir araçtır.
- Programcılar, bilgisayarları hayatımızın her alanında daha kullanışlı ve verimli hale getirirler.
- Otomasyon: Evlerimizde akıllı termostatlar, ışıklar ve güvenlik sistemleri.
- Sağlık: Tıbbi cihazlar ve hastane sistemleri.
- Eğitim: Online ders platformları ve öğretim yazılımları.
- Kendi fikirlerinizi gerçekleştirin: Bilgisayarlarınızı istediğiniz gibi programlayabilirsiniz.

Bilim Artık Hesaplamalı Veri İşlemeye Dönüşüyor

Eski Model - Dünyayı Sorgulama

- Eski bilim modeli, belirli bir hipoteze dayalı veri toplamayı içerir.
- Bilim insanları, dünyaya özel bir soru sorar ve buna yönelik veri toplarlar.

Yeni Model - Dünyayı İndirme

- Yeni bilim modeli, veri toplamanın birçok hipotezi desteklediğini gösteriyor.
- Bilim insanları, geniş veri kümesini indirir ve bu veriyi kullanarak farklı hipotezleri test ederler.

Bilimde Programlamanın Önemi

- Bilim, artık büyük verilerle daha yakından ilişkilidir.
- Programlama, bilimsel keşiflerin önemli bir aracıdır.
- Bilimde programlama, verileri anlama ve bilimsel keşifleri hızlandırma yeteneğini artırır.
- Bilim dünyasındaki teknolojik ilerlemelere katkı sağlar.
- Programlama, büyük veri kümesi ile çalışmayı kolaylaştırır.
- Bilim insanları, veriyi analiz etmek ve yeni hipotezleri test etmek için programlama becerilerini kullanır.

Uygulama Örnekleri

- Astronomi
  - Gözlem ve veri toplama: Yüksek çözünürlükte, yüksek frekansta gökyüzü taramaları (SDSS, LSST, PanSTARRS).
- Biyoloji
  - Laboratuvar otomasyonu: Yüksek kapasiteli dizileme (high-throughput sequencing).
- Deniz Bilimleri
  - Yüksek çözünürlüklü modeller: Uygun fiyatlı sensörler, uydu verileri, deniz altı keşifleri.

Hesaplama Ne Anlama Gelir?

- Hesaplama, bir bilgisayarın veya hesaplama cihazının bir görevi gerçekleştirmek için verileri işleme sürecidir.
- Bu işlem, matematiksel işlemler, mantıksal kararlar ve veri manipülasyonunu içerir.
- Hesaplama, bilgisayarlarımızın temel işlevlerinden biridir.
- Programlama, hesaplamayı daha etkili ve verimli hale getirme yeteneğini size kazandırabilir.
- Hesaplama, iş dünyasında, bilimde ve günlük yaşamda hayati bir rol oynar.
- Veri analizi, simülasyonlar, oyun geliştirme ve daha fazlası için kullanılır.

Hesaplamanın Yapı Taşları

- Matematik: Hesaplama temel olarak matematiksel işlemleri içerir. Toplama, çıkarma, çarpma, bölme vb.
- Mantık: Mantıksal ifadeleri değerlendirerek kararlar alırız. (Örnek: "Eğer hava güneşli ise dışarı çık.")
- Hesaplama, verileri alır, işler ve sonuçları üretir.
- Bu sonuçlar, sorunları çözmek, veri analizi yapmak veya görevleri otomatikleştirmek için kullanılabilir.
- Programlama, bilgisayarlarımıza hesaplama görevlerini nasıl yapacaklarını anlatmanın bir yoludur.
- Programcılar, algoritmaları kullanarak belirli bir işlemi gerçekleştirmek için kodlar yazarlar.

Bilgisayar Bilimi Sadece Teknoloji Öğrenmek Değildir

- Bilgisayar bilimi, sadece bilgisayarlarla ilgili teknik bilgi değil, aynı zamanda problem çözme becerileriyle ilgili bir disiplindir.
- Bilgisayar bilimi, bilginin nasıl işlendiği, depolandığı ve iletişim kurduğu hakkında derin bir anlayış gerektirir.
- Bilgisayar bilimi, sadece teknolojiyle sınırlı değildir. Teknoloji sadece bir araçtır.
- Temel amaç, sorunları çözmek ve bilgiyi daha iyi anlamaktır.
- Bilgisayar bilimi, karmaşık problemleri tanımlama, analiz etme ve etkili çözümler üretme yeteneklerini geliştirir.
- Bu beceriler, herhangi bir alanda başarılı olmanıza yardımcı olur.

Teknoloji ile Sosyal Sorumluluk

- Bilgisayar bilimi, teknolojinin insanlara nasıl yardımcı olabileceği ve toplumsal sorunların çözümüne nasıl katkı sağlayabileceği konularında düşünmeyi teşvik eder.
- Teknolojiyi insanlığın hizmetine sunma sorumluluğunu vurgular.
- Bilgisayar bilimi, tıp, ekonomi, astronomi, biyoloji gibi birçok farklı alanda uygulanabilir.
- Problemleri daha iyi anlamak ve çözmek için bilgisayar biliminin ilkelerini kullanabilirsiniz.
- Bilgisayar bilimi, sadece teknolojiyle ilgilenenler için değil, problem çözme ve düşünme becerilerini geliştirmek isteyen herkes için ilginç bir alandır.
- Bilgisayar bilimi, dünyayı daha iyi anlamamıza ve geliştirmemize yardımcı olabilir.

Bilgisayar Bilimi Nedir?

- Bilgisayar bilimi, sadece bilgisayarlarla ilgilenen bir alandır. Aynı zamanda mantık, problem çözme ve yaratıcılığa dayalı bir disiplindir.
- Bilgisayar bilimi, mantığı temel alır. Mantık, verileri değerlendirme ve mantıklı sonuçlar çıkarma yeteneğini içerir.
- Mantık, bilgisayar programlarını yazma sürecinin temelini oluşturur.
- Bilgisayar bilimi, karmaşık problemleri tanımlama ve çözme yeteneği gerektirir. Her problem, mantıklı bir adım sırasıyla çözülür.
- Programlama, problem çözme yeteneklerinizi geliştirmenize yardımcı olur.

Bilgisayar Bilimi Nedir?

- Bilgisayar bilimi, yaratıcı düşünmeyi teşvik eder. Programcılar, farklı yollarla aynı problemi çözmeyi düşünmelidir.
- Yaratıcılık, yeni ve yenilikçi çözümlerin doğmasına yol açar.
- Mantık, problem çözme ve yaratıcılık, herhangi bir alanda başarılı olmanıza yardımcı olur.
- Bilgisayar bilimi, sadece teknolojiyle ilgilenenler için değil, aynı zamanda düşünme ve yaratıcılık yeteneklerinizi geliştirmek isteyen herkes için ilginç bir alandır.

İlk Bilgisayar - Atanasoff-Berry Bilgisayarı (ABC)

- İlk bilgisayar olarak kabul edilen Atanasoff-Berry Bilgisayarı (ABC), John Atanasoff ve Clifford Berry tarafından 1930'ların sonlarında geliştirildi.
- ABC, temel mantık devreleri kullanarak dijital hesaplamaları yapabilen bir elektronik bilgisayardı.
- ABC, dijital bir sistemdi ve iki haneli ondalık sayıları işleyebiliyordu.
- Belleği, verileri ve komutları saklamak için kullanılan manyetik şeritlerden oluşuyordu.

İlk Program - Ada Lovelace ve Charles Babbage

- İlk program, Ada Lovelace ve Charles Babbage tarafından geliştirilen Analitik Makine için yazıldı.
- Ada Lovelace, Analitik Makine için bir dizi komut ve hesaplama yöntemi geliştirdi. Bu, ilk bilinen programdı.

İlk programlar

- İlk bilgisayarlar ve programlar, bilgisayar biliminin temelini attı.
- Günümüzde, programlama, dünyayı dönüştüren güçlü bir araç haline geldi.
- İlk programlar genellikle matematiksel hesaplamalarda kullanılıyordu.
- Bilgisayarların ilk kullanımları genellikle bilimsel ve askeri uygulamalara odaklanıyordu.
- İlk yüksek seviye programlama dili olan Fortran (Formula Translation), IBM tarafından geliştirildi ve 1957'de piyasaya sürüldü.
- Fortran, bilimsel hesaplamalarda yaygın olarak kullanılmıştır.

Bilgisayar Nedir?

- Bilgisayar, bir dizi hesaplama ve talimatı yürüten bir cihazdır.
- Modern bilgisayarlar elektronik ve dijitaldir, yani verileri elektronik olarak işlerler.
- Bilgisayarlar, verileri işlemek için matematiksel hesaplamaları ve mantıksal talimatları kullanır.
- Bu hesaplamalar ve talimatlar, bilgisayarın belirli bir görevi gerçekleştirmesini sağlar.
- Günümüzdeki bilgisayarlar, elektronik devrelerle çalışan ve sayıları dijital olarak temsil eden cihazlardır.
- İşlemciler, bellekler, depolama birimleri ve giriş/çıkış cihazları gibi bileşenler içerirler.

Elektronik ve Dijital

- Elektronik: Bilgisayarlar, elektrik akımı kullanarak verileri işler.
- Dijital: Verileri sıfırlar (0) ve birler (1) olarak temsil ederler.
- Bilgisayarlar, verileri işlemek ve sonuçları üretmek için programlar kullanır.
- Programlar, bilgisayarlara hangi hesaplamaları ve talimatları yapacaklarını söyler.
- Bilgisayarlar, modern dünyada hayati bir rol oynar.
- Programlama, bilgisayarların nasıl çalıştığını anlamamıza ve onları kullanarak sorunları çözmemize yardımcı olur.

Algoritmalar Nedir?

- Algoritma, adım adım bir problemi çözme prosedürüdür.
- Algoritmalar, bir görevi nasıl gerçekleştireceğimizi belirlememize yardımcı olur.
- Algoritmalar, bilgisayar programlarının temelini oluşturur.
- Bir problemi mantıklı ve etkili bir şekilde çözmek için kullanılırlar.
- Programlama, algoritmaları kullanarak bilgisayar programlarını oluşturma sürecidir.
- İyi bir programcı, sorunları çözmek için uygun algoritmaları seçebilir ve uygulayabilir.
- Programlar ve algoritmalar, bilgisayar dünyasının temel taşlarıdır.

Sabit Programlı Bilgisayarlar Nedir?

- Sabit programlı bilgisayarlar, belirli bir problem veya işlem kümesini çözmek için tasarlanmıştır.
- Bu tür bilgisayarlar, belirli bir görevi yapmak için önceden tanımlanmış bir programı çalıştırırlar.
- Sabit programlı bilgisayarların kökleri çok eskilere dayanır.
- İşte bazı örnekler: Abaküs, Antikythera Mekanizması, Pascaline, Leibniz Tekerleği, Jacquard'ın Dokuma Tezgahı, Babbage Fark Motoru, Hollerith Elektrik Tabulasyon Sistemi, Atanasoff-Berry Bilgisayarı (ABC), Turing Bombası, Ve daha fazlası...

Charles Babbage ve Ada Lovelace

- Charles Babbage, ilk programlanabilir bilgisayar olan Analitik Makine'nin tasarımını yapmıştır.
- Ada Lovelace, Analitik Makine için yazılmış ilk bilinen programdır. Bu nedenle bazen "ilk programcı" olarak kabul edilir.
- Bu sabit programlı bilgisayarlar, bilgisayar biliminin temellerini atmıştır.
- Teknolojik gelişmelerin ilham kaynağı olmuş ve bugünkü bilgisayarların öncüleri olarak kabul edilirler.
- Sabit programlı bilgisayarlar, bilgisayar teknolojisinin tarihinde önemli bir rol oynamıştır.
- Gelişmiş programlanabilir bilgisayarlar bugün hala hayatımızın vazgeçilmez bir parçasıdır.

Depolanan Programlı Bilgisayarlar Nedir?

- Depolanan programlı bilgisayarlar, belirli bir problemi çözmek için sabit bir program kullanmak yerine programlarını belleklerinde depolayabilen makinelerdir.
- Bu, daha esnek ve genel amaçlı bilgisayarların temelini atmıştır.
- Bu tür bilgisayarlar, problem çözmek için programlarını çalıştırabilirler. Bu programlar, görevin doğasına göre değiştirilebilir.
- Bu, bilgisayarların daha önce çözemediği problemleri ele alabilmesini sağlar.
- Düşünün ki, girişiniz başka bir makine veya bir makinenin tanımı olsun.
- Universal Turing makineleri, bu tür "makine girişi"ni işleyebilme yeteneğine sahiptir.

Evrensel Turing Makineleri

- Evrensel Turing makineleri, soyut bir genel amaçlı bilgisayardır.
- Herhangi bir Turing makinelerinin işlemesini simüle edebilirler ve bu nedenle her tür problemi çözebilirler.
- Evrensel Turing makineleri, bilgisayar biliminin temelini oluşturur.
- Alan Turing tarafından geliştirilen bu kavram, modern bilgisayarların evrensel programlanabilirliğinin temelini atmıştır.
- Depolanan programlı bilgisayarlar ve Evrensel Turing makineleri, bilgisayar teknolojisinin gelişiminde önemli bir dönüm noktasını temsil eder.
- Bu kavramlar, günümüzün genel amaçlı bilgisayarlarının temelini oluşturur.

Genel Amaçlı Bilgisayar Nedir?

- Genel amaçlı bir bilgisayar, farklı görevleri yerine getirebilen ve kullanıcıların ihtiyaçlarına göre programlanabilen bir cihazdır.
- Bu bilgisayarlar, geniş bir uygulama yelpazesi için tasarlanmıştır.
- Genel amaçlı bilgisayarlar, iş, eğitim, eğlence ve daha birçok alanda kullanılabilirler.
- Yazılım ve programlar değiştirilerek, farklı görevler için uyarlanabilirler.
- Bu bilgisayarlar, kullanıcıların ihtiyaçlarına göre programlanabilirler.
- Programlama, bilgisayara belirli bir görevi nasıl yerine getireceğini söyleme sürecidir.

Bilgisayarın İşlemesi

- Genel amaçlı bir bilgisayar, verileri işlemek için bir işlemci (CPU), bellek ve giriş/çıkış cihazları içerir.
- İşlemci, verileri programlara göre işler ve sonuçları üretir.
- Genel amaçlı bilgisayarlar, iş dünyasından bilimsel araştırmalara, eğitimden eğlenceye kadar birçok alanda kritik bir rol oynar.
- İş süreçlerini otomatikleştirir, bilimsel hesaplamaları hızlandırır ve iletişimi kolaylaştırır.
- Genel amaçlı bilgisayarlar, modern yaşamın ayrılmaz bir parçasıdır.
- Programlama, bu bilgisayarları istediğimiz gibi özelleştirmemize ve kullanmamıza olanak tanır.

Dijital Bilgisayarların Gücüne Sınırlar Var mı?

- Dijital bilgisayarlar, karmaşık hesaplamaları ve görevleri hızlı bir şekilde gerçekleştirebilirler.
- Bu güç, teknolojik gelişmelere dayalıdır.
- Bir bilgisayarın işlemci hızı, hesaplamaların ne kadar hızlı yapılacağını belirler.
- Ancak işlemci hızı sınırlıdır ve daha hızlı işlemciler yapmak her zaman mümkün değildir.
- Bellek, bilgisayarın verileri saklama kapasitesini temsil eder.
- Daha fazla bellek, daha büyük ve karmaşık görevleri yerine getirme yeteneğini artırabilir, ancak bu da sınırlıdır.

Dijital Bilgisayarların Gücüne Sınırlar Var mı?

- Bilgisayarların veri işleme kapasitesi sınırlıdır.
- Büyük veri kümelerini veya çok karmaşık hesaplamaları işlemek için zaman ve kaynak gerekebilir.
- Dijital bilgisayarlar, enerji tüketirler ve bu da sınırlı kaynakları etkiler.
- Daha güçlü bilgisayarlar, daha fazla enerjiye ihtiyaç duyarlar.
- Dijital bilgisayarlar inanılmaz derecede güçlüdür, ancak her zaman sınırları vardır.
- Programcılar, kaynakları etkin bir şekilde kullanarak bu sınırları aşmaya çalışırlar.

Makineler ve Teknoloji

- Teknoloji, insanların yaşamını büyük ölçüde etkileyen bir güçtür.
- Makineler, bu teknolojik gelişmenin önemli bir parçasını oluşturur.
- Makineler, işleri hızlı ve verimli bir şekilde yapabilme yeteneğine sahiptir.
- Bu güç, endüstriyel devrimden günümüze kadar büyük ölçüde artmıştır.
- Her şeye rağmen, inşa edebileceğimiz makinelerin de sınırları vardır.
- Bu sınırlar, fiziksel, enerji ve teknolojik kısıtlamalardan kaynaklanır.
- Fiziksel sınırlar, makinelerin boyutunu, ağırlığını ve dayanıklılığını etkiler.
- Bir şeyin ne kadar büyük veya küçük olabileceği bazen fiziksel yasalara bağlıdır.

Makineler ve Teknoloji

- Makineler enerjiye ihtiyaç duyarlar ve enerji kaynakları sınırlıdır.
- Daha güçlü makineler, daha fazla enerjiye ihtiyaç duyarlar.
- Teknolojik gelişmeler, makinelerin gücünü artırabilir.
- Yeni malzemeler, tasarım fikirleri ve algoritmalar, sınırları zorlayabilir.
- Makineler ve teknoloji, insan hayatını dönüştürdü ve daha da dönüştürecek.
- Sınırları anlamak, daha iyi ve daha etkili makineler tasarlamamıza yardımcı olabilir.

Evrensel Turing Makineleri

- Universal Turing makineleri, bilgisayar biliminin en temel soyut modellerinden biridir.
- Herhangi bir Turing makinesinin işlemlerini simüle edebilme yeteneğine sahiptirler.
- Bant (Tape)
  - Universal Turing makineleri işlem için bir bant kullanır.
  - Bant, giriş verisini, çıkışları ve ara sonuçları saklar.
  - Bant, birbirine bağlı hücrelerden oluşur ve her hücreye bir sembol yazılabilir.

Evrensel Turing Makineleri

- Bant Başlığı (Tape Head)
  - Bant başlığı, bant üzerindeki bir hücreyi işaret eder.
  - Aktif hücredeki sembolü okur, yeni bir sembol yazar ve bir hücre ileri veya geri hareket edebilir.
- Hücreler ve Semboller
  - Bant hücreleri, birer sembol içerebilir. Bu semboller, işlemi yönlendirmek için kullanılır.
  - Bir hücredeki sembol, Turing makinesinin mevcut durumunu ve hangi işlemi yapması gerektiğini belirler.

Evrensel Turing Makineleri

- Universal Turing makineleri, hesaplamaların sınırlarını tanımlar.
- Bu model, diğer tüm bilgisayarlarla eşdeğerdir ve hesaplanabilir her şeyi hesaplayabilir.
- Universal Turing makineleri kavramı, 20. yüzyılın en önemli bilimsel sonuçlarından biri olarak kabul edilir.
- Bu kavram, bilgisayar biliminin temellerini atmıştır ve modern bilgisayarların çalışma mantığının anlaşılmasına katkıda bulunmuştur.
- Universal Turing makineleri, bilgisayar biliminin temelini oluşturan soyut bir modeldir.
- Bu model, hesaplama ve problem çözme süreçlerinin sınırlarını anlamamıza yardımcı olur.

Turing Makineleri

- Turing makineleri, herhangi bir doğal dünya işlemi tarafından hesaplanabilen işlevleri hesaplayabilir.
- Bu, bilgisayar biliminin temel teoremlerinden biridir.
- Doğal Dünya İşlemleri
  - Doğal dünya işlemleri, fiziksel olayları ve süreçleri ifade eder.
  - Turing makineleri, bu tür işlemleri modellemek ve hesaplamak için kullanılabilir.
- Fiziksel Olarak Yararlanabilir İşlemler
  - "Fiziksel olarak yararlanabilir işlemler," doğal dünyanın gerçek fiziksel süreçlerini temsil eder.
  - Turing makineleri, bu tür işlemleri simüle edebilme yeteneğine sahiptirler.

Ada Lovelace Kimdir?

- Ada Lovelace, 19. yüzyılın ilk yarısında yaşamış bir İngiliz matematikçi ve yazardır.
- Ada, bilgisayar programlamasının öncüsü olarak kabul edilir.
- Ada Lovelace, Analitik Makine için yazdığı algoritma ve programlarıyla bilgisayar tarihindeki ilk programcılardan biri olarak kabul edilir.
- Ada Lovelace, bilimsel bir yaklaşımla matematiksel düşünceyi bilgisayar programlamasıyla birleştirmiştir.
- Ada Lovelace, bilgisayar biliminin öncülerinden biridir ve bilgisayar programlamasının temelini atmıştır.
- Onun azmi ve katkıları, bugün bile programlamaya ilgi duyanları ilham verir.

Babbage'ın Analitik Makinesi

- Charles Babbage, 19. yüzyılda yaşamış İngiliz bir matematikçi, mucit ve bilgisayar bilimcisidir.
- Analitik Makine, Charles Babbage tarafından tasarlanan ve geliştirilen bir mekanik bilgisayardır.
- Analitik Makine, modern bilgisayarların öncüsü olarak kabul edilir.
- Analitik Makine, matematiksel hesaplamalar yapabilen ve sonuçları saklayabilen bir cihazdı.
- Analitik Makine, farklı hesaplamaları yapmak için programlanabilirdi.
- Analitik Makine, modern bilgisayarların temelini atmıştır.
- Bu makinenin tasarımı ve fikirleri, günümüzün bilgisayarlarının çalışma mantığına benzerlik gösterir.
- Bu icat, bilgisayarların gelişimine büyük bir katkıda bulunmuştur.

Colossus Mark 1

- Colossus Mark 1, 1944 yılında Birleşik Krallık'ta inşa edilen ve programlanabilir ilk elektronik dijital bilgisayardır.
- İkinci Dünya Savaşı sırasında, düşmanın iletişimini çözmek için gizli kodlar kullanılıyordu. Colossus, bu kodları çözmek için kullanılan bir araçtı ve savaşın sonucunu etkiledi.
- Colossus, elektronik bir bilgisayar olarak kabul edilir çünkü elektronik valfler (vakuüm tüpleri) kullanıyordu.
- Bu, mekanik cihazlara göre çok daha hızlı ve esnek bir işlem yapma yeteneği sağladı.
- Colossus, programlanabilir bir bilgisayardı, yani işlevleri değiştirilebilirdi.
- İlk programlanabilir elektronik dijital bilgisayar olarak kabul edilir.
- Colossus Mark 1, sadece savaş sırasında değil, aynı zamanda bilgisayar teknolojisinin ilerlemesine de büyük katkıda bulundu.

ENIAC

- ENIAC, 1946 yılında Amerika Birleşik Devletleri'nde John Mauchly ve J. Presper Eckert tarafından geliştirilen bir bilgisayardır.
- ENIAC, genel amaçlı bir elektronik bilgisayardı, yani farklı işlemleri gerçekleştirebilecek şekilde tasarlanmıştı.
- ENIAC, mekanik parçalar yerine elektronik valfler (vakuüm tüpleri) kullanıyordu.
- Bu, daha hızlı ve daha güvenilir bir işlem yapma yeteneği sağladı.
- ENIAC, büyük ölçekli ve hızlı hesaplamalar için kullanılabiliyordu.
- ENIAC, o dönemin en büyük bilgisayarlarından biriydi ve bu döneme ait görsel bir şölen sunuyor.
- İlk büyük ölçekli elektronik bilgisayar olarak kabul edilir.

EDVAC

- EDVAC, 1951 yılında Amerika Birleşik Devletleri'nde John von Neumann tarafından geliştirilen bir bilgisayardır.
- ENIAC'tan farklı olarak, EDVAC ikili sayı sistemi kullanıyordu. Bu, bilgisayarlar arasında evrensel bir standarttır.
- İkili sistem, yalnızca 0 ve 1'leri kullanarak bilgiyi temsil eder.
- EDVAC, programları ve verileri bellekte sıralı olarak depoluyordu.
- Bu, daha karmaşık ve esnek programların yazılabilmesini sağladı.
- Komutlar, sıralı olarak çalışırdı, ancak koşullu bir komut atlama yapabilir ve başka bir yere gidebilirdi.
- Von Neumann mimarisi, modern bilgisayarların temelini atmıştır.
- EDVAC'ın tasarımı, günümüz bilgisayarlarının temel çalışma mantığını yansıtır.

Bilgi Nedir?

- Bilgi, insanların veya makinelerin dünyayı anlamalarına ve çeşitli sorunları çözmelerine yardımcı olan bilinçli anlayışı ifade eder.
- Bilgi, çeşitli şekillerde ifade edilebilir.
  - Bildirimsel Bilgi (declarative)
  - Buyurusal Bilgi (imperative)

Bildirimsel Bilgi

- Deklaratif bilgi, ifade edilen veya açıkça ifade edilen bilgi türüdür. Bu tür bilgi, doğru veya yanlış olarak doğrulanabilir.
- Deklaratif bilgi, aksiyomlar veya tanımlar yoluyla ifade edilebilir.
- Aksiyomlar, bir şeyin ne olduğunu veya nasıl tanımlandığını belirtir.
- Deklaratif bilgi aynı zamanda gerçeklerin ifadesini içerebilir.
- Gerçekler, dünya hakkında doğru ve doğrulanabilir ifadelerdir.
- Örnek Bir Aksiyom
  - Örneğin, "y, x'in karekköüdür, yalnızca y*y = x ise" şeklinde bir aksiyom açıkça bir tanımlamayı ifade eder.
- Ancak bu aksiyom, karekökü hesaplamak için nasıl yapılacağı hakkında bilgi vermez.

Buyurusal Bilgi

- İmperatif bilgi, "nasıl bir şey yapılacağı" konusunda talimatlar içerir. Bu, özel bir görevin veya işlemin adımlarını içerir.
- Babylonian yöntemi, bir sayının karekökünü hesaplama işlemi için bir örnektir. Bu yöntem, "nasıl yapılacağı" hakkında talimatlar sunar.
- Yöntemin Adımları
  - İşlem, x'i bir girdi olarak alır.
  - İlk adım, bir başlangıç değeri olan y0 ile başlar.
  - Eğer yn^2 ≈ x ise, işlem sona erer.
  - Değilse, yn+1 = (yn + x/yn)/2 ile yeni bir değer hesaplanır.
  - Adım (3)'ü tekrarla.
- Bu yöntem, bir başlangıç tahminiyle başlar. Bu tahmin, karekökün yaklaşık değerini verir.
- Adımlar, yaklaşık sonuca daha fazla yaklaşmak için tekrarlanır.

En Büyük Ortak Bölgenin (GCD) Tahmin Edilmesi

- GCD, iki veya daha fazla sayının en büyük ortak bölenini ifade eder.
- İki sayının GCD'si, bu sayılara tam olarak bölünebilen en büyük pozitif tam sayıdır.
- Deklaratif Tanım
  - Deklaratif bir tanım, GCD'nin ne olduğunu açıklar.
  - "d, a ve b'nin GCD'si ise, d a = dx ve b = dy eşitliklerini sağlayan en büyük pozitif tam sayıdır."
- İmperatif Tanım: Öklidyen Algoritması
  - İki sayının GCD'sini bulmak için kullanılan yaygın bir algoritmadır.
  - İmperatif bir yaklaşım sunar ve nasıl yapılacağına dair talimatlar içerir.

Öklidyen Algoritması

- Algoritmanın Adımları
  - İşlem, iki pozitif tam sayı olan a ve b'yi alır (a >= b).
  - a'yı b'ye böl ve kalanı R olarak adlandır.
  - Eğer R = 0 ise, işlem sona erer.
  - Aksi takdirde, a'yı b'ye, b'yi de R'ye eşitler.
  - Adım (3)'ü tekrarla.

Bilgisayar Biliminde Neler Var?

- Bilgisayar bilimi, soyutlama, problem çözme, yaratıcılık ve bilimsel keşiflerin birleşimini temsil eder.
- Bilgisayar biliminde soyutlama, karmaşık bir sistemi veya problemi basitleştirmek ve anlamak için kullanılan bir kavramdır. Bilgisayarlar soyutlama yeteneği sayesinde karmaşık görevleri gerçekleştirir.
- Bilgisayar bilimi, temelde problemleri çözme sanatıdır. Programcılar, sorunları çözmek için algoritmalar ve kod kullanır.
- Bilgisayar bilimi, sadece mantık ve analitik düşünceyle sınırlı değildir. Sanat, yaratıcılık ve tasarım da bu alanda büyük bir rol oynar. Örneğin, dijital medya, elektronik müzik, oyunlar ve animasyon gibi alanlarda sanatsal ifadeye imkan tanır.
- Bilgisayar bilimi aynı zamanda bir bilim dalıdır. Gerçekliği anlama, modelleme ve analiz etme amacı güder. Bilgisayarlar, büyük veri analizi ve simülasyonlar gibi bilimsel araştırmalarda önemli bir rol oynar.
- Bilgisayar bilimi, farklı kariyer olanakları sunar. Yazılım geliştirme, veri bilimi, siber güvenlik, yapay zeka, oyun geliştirme gibi birçok alanda çalışma fırsatları vardır.

Algoritmalar: Ana Düşünce

- Bir algoritma, bir sorunu çözmek için izlenmesi gereken adımları belirleyen bir tariftir.
- Algoritmalar, karmaşık sorunları basitleştirmek ve çözmek için kullanılır.
- Bilgisayar bilimi, geniş bir konsept olarak kabul edilebilir, ancak temelde algoritmaların çalışma ve analizini içerir.
- Algoritmaların tasarımı ve uygulanması bilgisayar biliminin merkezindedir.
- Algoritmalar, hayatımızın birçok yönünü etkiler. İnternet aramalarından yol tariflerine kadar birçok günlük görev, algoritmaların yardımıyla çalışır.
- Algoritmalar, basit adımların belirlenmesi, mantıklı bir sıra oluşturulması ve problemi çözme amacı güder.
- İyi tasarlanmış algoritmalar, verimlilik ve doğruluk açısından önemlidir.
- Algoritmaların anlaşılması ve tasarlanması, bilgisayar biliminin merkezindeki önemli bir beceridir.

Algoritmalar: Özet

- Bir algoritma, bir problemi çözmek için izlenmesi gereken adımları belirleyen bir reçetedir.
- Algoritmalar, karmaşık problemleri basitleştirmek ve çözmek için kullanılır.
- Bilgisayar bilimi, geniş bir alandır, ancak temelde algoritmaların çalışma ve analizini içerir.
- Bilgisayar bilimi, otomatik yöntemlerle problemleri çözmenin çalışmasıdır.
- İyi bir algoritma, basit adımların belirlenmesi, mantıklı bir sıra oluşturulması ve problemi çözme amacı güder.
- Algoritmaların tasarımı, verimlilik ve doğruluk açısından kritik öneme sahiptir.
- Bilgisayar bilimi, temelde otomatik yöntemlerle problemleri çözmenin çalışmasıdır.
- Programlar, algoritmaların uygulanmasının bir yoludur.

Problem Belirleme

- Herhangi bir işin başarılı bir şekilde yapılabilmesi için, sorunun ne olduğunun önceden belirlenmesi gerekir.
- Problem belirleme, bir sorunun doğru bir şekilde tanımlanmasını ve anlaşılmasını sağlar.
- Algoritmalar, bu belirlenen problemleri çözmek için kullanılır ve sorunların çözümünü otomatikleştirir.
- Bir belirleme veya özellikle bir problem belirleme, bir sorunun ne olduğunu ve ne tür bir sonuca ulaşılması gerektiğini belirler.
- Problem belirleme, problemin boyutunu ve karmaşıklığını anlamamıza yardımcı olur.
- Algoritmalar, belirlenen problemleri çözmek için kullanılır.
- Bir algoritma, problemi adım adım çözmek için izlenmesi gereken bir yol haritasıdır.

Problem Belirleme

- Problem belirleme, bir programın veya algoritmanın ne yapması gerektiğini açıkça tanımlama sürecidir.
- Bu, bir problemi çözmek veya bir görevi yerine getirmek için gerekli olan adımları ve çıktıyı tanımlamayı içerir.
- Bir problem belirlerken, genellikle iki temel bileşen vardır: girdi (INPUT) ve çıktı (OUTPUT).
  - Girdi, problemin çalışması için gereken bilgi veya verileri temsil eder.
  - Çıktı, problemin sonucunu veya istenen bilgiyi temsil eder.

Örnek: İki Sayının Toplamı

- Örnek bir problem belirlemeye bakalım:
  - GİRDİ: İki sayı, X ve Y.
  - ÇIKTI: X ve Y'nin toplamı olan tek bir sayı, Z (Z = X + Y).
- Örnek Sorun Açıklaması
  - Problem: Verilen iki sayıyı topla.
  - GİRDİ: İki sayı, X ve Y.
  - ÇIKTI: Bu iki sayının toplamı olan tek bir sayı, Z.
- Eğer X = 5 ve Y = 3 ise, Z = 8 olmalıdır.

Problem Belirleme Örnekleri

- Örnek: Hisse Senedi Piyasası Tahminleri
  - GİRDİ: Hisse senedi piyasasından gelen veriler.
  - ÇIKTI: Piyasa hakkında doğru tahminler.
- Problem Açıklaması
  - Problem: Hisse senedi piyasasının gelecekteki durumunu tahmin etmek.
  - GİRDİ: Hisse senedi piyasasından gelen veriler, fiyatlar, hacimler vb.
  - ÇIKTI: Piyasa hakkında doğru tahminler, yani hisse senetlerinin gelecekteki performansı hakkında bilgi.
- Eğer girdi olarak son bir yılın hisse senedi fiyatları ve ekonomik veriler verilirse, çıktı olarak gelecek bir yılın piyasa trendi tahmin edilmelidir.

Pseudocode Kullanımı

- Algoritma, matematiksel bir problemi çözmek için sınırlı sayıda adım içeren ve genellikle bir işlemi tekrarlama gerektiren bir prosedürdür.
- Algoritma, belirli bir görevi yerine getirmek için adım adım bir yöntemdir.
- Algoritmalar genellikle "pseudocode" kullanılarak ifade edilir.
- Algoritmalar, belirli bir problemi çözme veya bir görevi yerine getirme amacı güder.
- Adım adım ilerler ve her adımda belirli bir işlem gerçekleştirir.
- Genellikle tekrarlanabilir işlemleri içerir.

Pseudocode Kullanımı

- Algoritmalar genellikle insanlar tarafından daha kolay anlaşılabilmesi için "pseudocode" kullanılarak ifade edilir.
- Pseudocode, insan diline benzer bir dil kullanır ve algoritmanın mantığını açıklar.
- Örnek: Toplama İşlemi
  - Problem: İki sayının toplamını hesapla.
  - Algoritma:
    - İlk sayıyı al (A).
    - İkinci sayıyı al (B).
    - A ve B'yi topla.
    - Sonucu çıktı olarak ver.

Pseudo Kod Nedir?

- Pseudo kod, İngilizce (veya herhangi bir dillerde) ifadeleri, genellikle programlama dillerinin ifadelerine benzeyen bir yapıya dönüştüren bir metottur.
- Adımlar, genellikle numaralandırılmış ve girintili olarak sunulur.
- Çoğu işlem için sabit bir sözdizimi gerekmez.
- Doğal dilden daha az belirsiz ve daha okunaklıdır.
- Pseudo kodun kullanılmasının nedenleri şunlar olabilir:
  - İşlemi daha anlaşılır hale getirir.
  - Mantık hatalarını tespit etmeyi kolaylaştırır.
  - Programcılar arasında işbirliğini kolaylaştırır.
  - Algoritma davranışı hakkında mantıklı çıkarlara izin verir.
  - Programlama diliyle kod yazma aşamasına geçiş yapmayı kolaylaştırır.

Algoritmik İşlem Türleri

- Algoritmalar, sıralı, koşullu ve döngüsel işlemleri kullanarak işleri gerçekleştirir.
- Bu işlem türleri, her programlama dilinde bulunur ve temel programlama becerilerinin bir parçasıdır.
- Bu işlem türleri, algoritmaların temel yapısını oluşturur.
- Her programlama dilinde bu işlemleri uygulayabiliriz.
- İşlem sıraları, koşullar ve döngüler, karmaşıklığı azaltmak ve işlemleri kontrol etmek için kullanılır.

Algoritmik İşlem Türleri

- Sıralı İşlemler (Sequential Operations)
  - Sıralı işlemler, adım adım sırayla gerçekleştirilen işlemlerdir.
  - Her işlem bir öncekinden sonra gelir ve sırayla çalışır. Örnek: Verilerin okunması, işlenmesi ve sonuçların üretilmesi.
- Koşullu İşlemler (Conditional Operations)
  - Koşullu işlemler, bir koşulun karşılanıp karşılanmadığını kontrol ederek işlem akışını değiştirir.
  - Eğer belirli bir koşul sağlanıyorsa, belirli bir işlem yapılır; aksi halde başka bir işlem yapılır. Örnek: Bir sayının pozitif veya negatif olduğunu kontrol etme.
- Döngüsel İşlemler (Iterative Operations)
  - Döngüsel işlemler, belirli bir koşul karşılanana kadar belirli bir işlemi tekrarlar.
  - Bir işlem veya işlemler kümesi belirli bir koşula uygun olana kadar çalışmaya devam eder. Örnek: Bir dizi elemanın tümünün toplanması.

Sıralı İşlemler: Giriş, Hesaplama ve Çıkış

- Sıralı işlemler, bir programın adım adım sırayla gerçekleştirilen işlemlerini ifade eder.
- Giriş İşlemleri
  - Giriş işlemleri, dış dünyadan veri değerlerini almak için kullanılır.
  - Örnek: Bir kişinin ağırlığını (w) ve boyunu (h) almak.
- Hesaplama İşlemleri
  - Hesaplama işlemleri, değişkenlerin değerlerini ayarlamak ve aritmetik işlemler yapmak için kullanılır.
  - Değişken, veri değerini tutabilen isimlendirilmiş bir depolama konumudur.
  - Örnek: BMI (Vücut Kitle İndeksi) hesaplamak için h / (w * w) işlemi.
- Çıkış İşlemleri
  - Çıkış işlemleri, sonuçları dış dünyaya göndermek ve görüntülemek için kullanılır.
  - Örnek: BMI'nin (vücut kitle indeksi) değerini yazdırmak.

Örnek - İki Sayıyı Toplama

- Bu örnek, iki sayının toplama işlemi adım adım nasıl gerçekleştirildiğini göstermektedir.
- Adım 1: Başla
  - İşlemimize "Başla" adımıyla başlıyoruz.
- Adım 2: Birinci Sayıyı Al
  - İlk olarak, birinci sayıyı (a) alıyoruz. Bu kullanıcının girdisi olabilir.
- Adım 3: İkinci Sayıyı Al
  - Ardından, ikinci sayıyı (b) alıyoruz. Bu da kullanıcının girdisi olabilir.
- Adım 4: c'yi Hesapla
  - Şimdi, c'yi hesaplamak için bir işlem yapma zamanı geldi. c, a ve b'nin toplamıdır.
- Adım 5: c'yi Yazdır
  - Hesaplanan c değerini ekrana yazdırıyoruz. Bu, sonucun görüntülenmesini sağlar.
- Adım 6: Bitir
  - Ve son olarak, işlemimizi "Bitir" adımıyla tamamlıyoruz.

Örnek - Dikdörtgenin Alanı

- Bu örnek, bir dikdörtgenin alanının hesaplanmasını adım adım göstermektedir.
- Adım 1: Başla
  - İşlemimize "Başla" adımıyla başlıyoruz.
- Adım 2: Dikdörtgenin Tabanını Al
  - İlk olarak, dikdörtgenin taban uzunluğunu (b) alıyoruz. Bu kullanıcının girdisi olabilir.
- Adım 3: Dikdörtgenin Yüksekliğini Al
  - Ardından, dikdörtgenin yüksekliğini (h) alıyoruz. Bu da kullanıcının girdisi olabilir.
- Adım 4: Alanı Hesapla
  - Şimdi, dikdörtgenin alanını hesaplamak için bir işlem yapıyoruz. Alan, taban (b) ile yükseklik (h) çarpımına eşittir.
- Adım 5: Alanı Yazdır
  - Hesaplanan alanı ekrana yazdırıyoruz. Bu, sonucun görüntülenmesini sağlar.
- Adım 6: Bitir
  - Ve son olarak, işlemimizi "Bitir" adımıyla tamamlıyoruz.

Koşullu İşlemler: Soru Sor ve Alternatif Eylemler Seç

- Koşullu işlemler, bir programın sorular sorduğu ve cevaplara göre farklı eylemler seçtiği durumları ele alır.
- Koşullu işlemler, genellikle "eğer" bir şart sağlanıyorsa şunu yap, aksi takdirde başka bir şey yap şeklinde ifade edilir.
- İşte bir örnek: Eğer x 100'den büyükse, x'i yazdır; aksi takdirde x'e 100 ekleyin.
- Koşullu işlemler daha karmaşık sorular sormamızı sağlar. Cevaplar genellikle mantıklı (Doğru veya Yanlış) olur.
- Daha karmaşık bir örnek: Eğer x 100'den büyükse ve y 200'e eşitse, o zaman...
- Koşullu işlemler, programların çeşitli durumlar arasında geçiş yapmasını sağlar.
- Programların belirli koşullara göre nasıl davranacağını kontrol etmekte kullanılır.
- Mantıklı ifadeler (Doğru veya Yanlış) kullanılarak kontrol edilirler.

Örnek - İki Sayıyı Al, Daha Büyüğünü Yazdır

- Bu örnek, iki sayı alınır ve daha büyük olanı belirlemek için bir koşullu işlem kullanır.
- Adım 1: Başla
  - İşlemimize "Başla" adımıyla başlıyoruz.
- Adım 2: İlk Sayıyı Al
  - İlk olarak, ilk sayıyı (f) alıyoruz. Bu kullanıcının girdisi olabilir.
- Adım 3: İkinci Sayıyı Al
  - Ardından, ikinci sayıyı (s) alıyoruz. Bu da kullanıcının girdisi olabilir.
- Adım 4: Daha Büyükse İlk Sayıyı Yazdır
  - Şimdi, bir koşul belirliyoruz: Eğer ilk sayı (f), ikinci sayıdan (s) büyükse, o zaman ilk sayıyı (f) yazdır.
- Adım 5: Aksi Takdirde İkinci Sayıyı Yazdır
  - Aksi takdirde (yani, f <= s), ikinci sayıyı (s) yazdırıyoruz.
- Adım 6: Bitir
  - Ve son olarak, işlemimizi "Bitir" adımıyla tamamlıyoruz.

İteratif İşlemler: Tekrar Eden Eylemler

- İteratif işlemler, belirli bir devam koşulu yanlış olana kadar belirli eylemleri tekrar etme yeteneğini ifade eder.
- İteratif işlemler, sıklıkla bir eylem kümesinin belirli bir koşul sağlanana kadar tekrarlanmasını içerir.
- İteratif işlemler, belirli bir işlemi tekrar etmek veya döngü içinde çalışmak için kullanılır.
- Verileri işlemek, listeleri gezmek ve çok daha fazlasını yapmak için kullanışlıdır.

İteratif İşlemler: Tekrar Eden Eylemler

- İşte bir örnek: Eğer j 0'dan büyükse, s'yi s + aj olarak ayarla ve j'yi 1 azalt. Bu işlem, j 0'dan büyük olduğu sürece tekrarlanır.

```
while j > 0 do
        set s to s + aj
        set j to j - 1
```

- Başka bir örnek: Bir işlemi n değerinden büyük olana kadar tekrarlayın. Bu, belirli bir koşul sağlandığında döngünün sona ereceği "tekrarla" işlemi içerir.

```
repeat
        print ak
        set k to k + 1
until k > n
```

Örnek - Bir Listede En Büyüğünü Bulma

- Bu örnek, bir listenin içinde dolaşarak en büyük pozitif sayıyı bulma işlemini adım adım göstermektedir.
- Adım 1: Başla
  - İşlemimize "Başla" adımıyla başlıyoruz.
- Adım 2: En Büyük Başlangıçta 0 Olarak Ayarlanır
  - Bu, şu an için elimizde en büyük sayının olmadığını gösterir.
- Adım 3: Kontrol Edilecek Bir Öğe Var mı?
  - Listenin içinde kontrol edilecek bir öğe var mı diye bakıyoruz. Eğer varsa, devam ederiz.
- Adım 3: Öğe En Büyükse
  - Eğer kontrol ettiğimiz öğe, şu ana kadar bulduğumuz en büyük sayıdan (largest) büyükse, en büyük sayıyı bu öğe olarak güncelliyoruz.
- Adım 4: En Büyük Sayıyı Yazdır
  - Her döngü dönüşünde, şu ana kadar bulduğumuz en büyük sayıyı ekrana yazdırıyoruz.
- Adım 5: Bitir
  - Ve son olarak, işlemimizi "Bitir" adımıyla tamamlıyoruz.

Koşullu ve İteratif İşlemler

- Koşullu ve iteratif işlemler, programlarınızın belirli koşullara ve tekrarlara göre nasıl davranacağını kontrol etmenizi sağlar.
- Koşullu işlemler, belirli koşullara göre eylemleri kontrol eder.
- İteratif işlemler, belirli koşullara bağlı olarak eylemleri tekrarlar.
- Koşullu ve iteratif işlemler, programlarınızın daha esnek ve güçlü olmasını sağlar.
- Doğru koşullar ve iyi tasarlanmış döngüler, programlarınızın beklenen şekilde çalışmasını sağlar.
- Sonsuz döngüler hatalara neden olabilir, bu nedenle devam koşulları dikkatle ayarlanmalıdır.

Koşullu ve İteratif İşlemler

- Bir döngünün iki temel bileşeni vardır:
  - Devam Koşulu (Continuation Condition): Döngünün ne zaman sona erdiğini belirler. Bu koşul yanlış (false) olduğunda döngü sona erer.
  - Döngü Gövdesi (Loop Body): Döngünün her tekrarında gerçekleştirilen eylemleri içerir.
- Sonsuz Döngü
  - Sonsuz döngü, devam koşulu hiçbir zaman yanlış (false) olmayan bir döngü türüdür.
  - Bu, bir programın çalışmasını durduramayacağınız bir hataya neden olabilir.
- Sonsuz Döngü Hatası
  - Sonsuz döngü hatası, programlarınızı çökertecek veya yanıt vermeyen bir duruma getirebilecek bir hatadır.
  - Devam koşulu düzgün bir şekilde ayarlanmazsa bu tür hatalar ortaya çıkabilir.

SON