



Bölüm 1: Hesaplama ve Hesaplamalı Düşünme

JAVA ile Nesne Yönelimli Programlama



Programlama

- Bilgisayara talimatlar vermenin bir yoludur.
- Bilgisayarlar programlanarak karmaşık görevleri yerine getirebilirler.
- Problem çözme ve zihinsel yetenekleri geliştirir.
- Programlama dilleri, bilgisayarla iletişim kurmayı sağlayan araçlardır.



Bilgisayarlar Artık Her Yerde!

- Evlerden iş yerlerine, eğlenceden eğitime kadar her yerde kullanılır.
- Akıllı telefonlar, tabletler, dizüstü bilgisayarlar, ev aletleri, arabalar ..
- Bilgisayarlar hayatımızın vazgeçilmez bir parçası haline gelmiştir.
- İnternet ve dijital teknolojiler sayesinde dünya bir tık uzaklıkta.
- Alışveriş, haberler, iletişim, eğitim: hepsi dijital dünyada.



Programlamanın Rolü

- Programlama, bilgisayarların istediğimiz görevleri yerine getirmesi için kullanılan bir araçtır.
- Programcılar, bilgisayarları hayatımızın her alanında daha kullanışlı ve verimli hale getirirler.
 - **Otomasyon:** Akıllı termostatlar, ışıklar ve güvenlik sistemleri.
 - **Sağlık:** Tıbbi cihazlar ve hastane sistemleri.
 - **Eğitim:** Online ders platformları ve öğretim yazılımları.



Hesaplama

- Bilgisayarların temel işlevlerinden biridir.
- Bilgisayarın veya hesaplama cihazının bir görevi gerçekleştirmek için verileri işleme sürecidir.
- Matematiksel işlemler, mantıksal kararlar ve veri işlemeyi içerir.
- Programlama, hesaplamayı daha etkili ve verimli hale getirir.
- 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:** Kararlar mantıksal ifadeler değerlendirilerek alınır.
 - Örnek: "Eğer hava güneşli ise dışarı çık."
- Hesaplama, verileri alır, işler ve bir sonuç üretir.
- Bu sonuçlar, sorunları çözmek, veri analizi yapmak veya görevleri otomatikleştirmek için kullanılabilir.
- Programlama, bilgisayara hesaplamayı nasıl yapacağını anlatır.



Bilgisayar Bilimi Sadece Teknoloji Öğrenmek Değildir

- Bilgisayar bilimi, sadece bilgisayarlarla ilgili teknik bilgi değil, aynı zamanda problem çözme becerileriyle ilgili bir disiplindir.
- Bilginin nasıl işlendiği, depolandığı ve iletişim kurduğu hakkında derinlemesine anlayış gerektirir.
- 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.



Teknoloji ile Sosyal Sorumluluk

- Bilgisayar bilimi, teknolojinin insanlara nasıl yardımcı olabileceği ve toplumsal sorunların çözümüne nasıl katkı sağlayabileceği konularında düşünmeyi teşvik eder.
- Dünyayı daha iyi anlamamıza ve geliştirmemize yardımcı olabilir.
- Teknolojiyi insanlığın hizmetine sunma sorumluluğunu vurgular.
- Sadece teknolojiyle ilgilenenler için değil, problem çözme ve düşünme becerilerini geliştirmek isteyenler için de ilginç bir alandır.
- Tıp, ekonomi, astronomi, biyoloji gibi birçok farklı alanda uygulanabilir.



Bilgisayar Bilimi

- Mantık, problem çözme ve yaratıcılığa dayalı bir disiplindir.
- 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.
- Yaratıcılık, yeni ve yenilikçi çözümlerin doğmasına yol açar.
- Programlama, problem çözme yeteneğini geliştirmeye yardımcı olur.



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.
- Temel mantık devreleri kullanarak dijital hesaplamaları yapabilen elektronik bir bilgisayardı.
- Sayısal (*digital*) 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.





İlk Programlar

- Genellikle matematiksel hesaplamalarda kullanılıyordu.
- Bilgisayarların ilk kullanımları bilimsel/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.





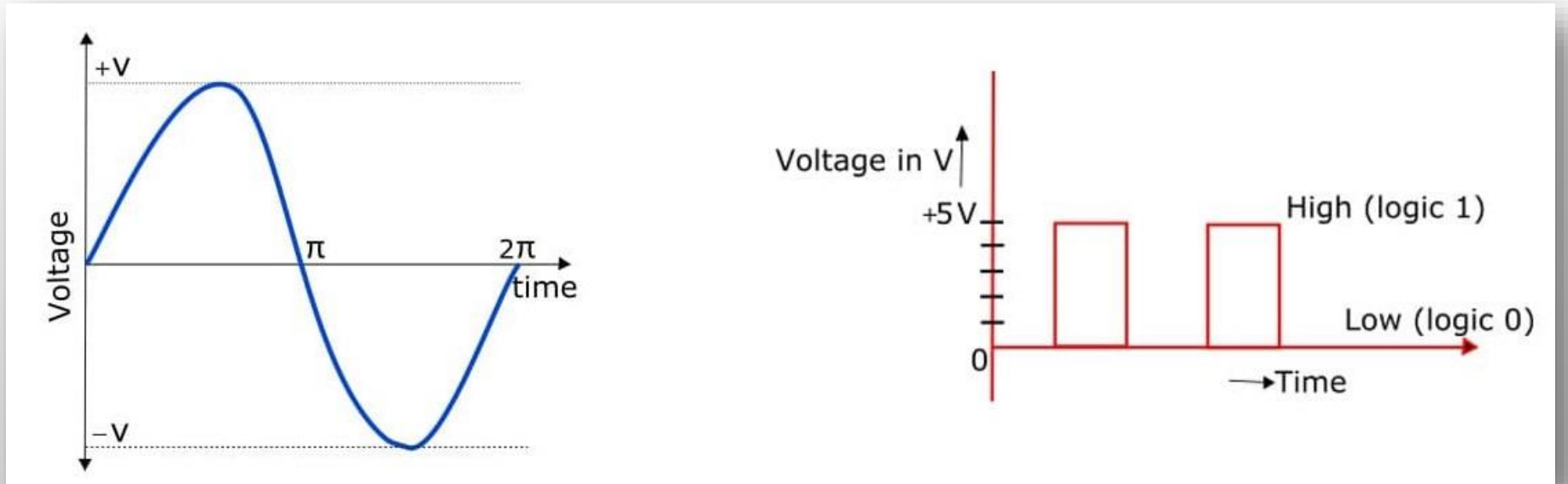
Bilgisayar

- Bir dizi hesaplama ve talimatı yürüten cihazdır.
- Modern bilgisayarlar elektronik ve dijitaldir, verileri elektronik olarak işlerler.
- Verileri işlemek için matematiksel hesaplamaları ve mantıksal talimatları kullanır.
- Hesaplama ve talimatlar, bilgisayarın bir görevi gerçekleştirmesini sağlar.
- Günümüzdeki bilgisayarlar, elektronik devrelerle çalışan ve sayıları dijital olarak temsil eden cihazlardır.
- İşlemci, bellek, depolama birimi ve giriş/çıkış cihazı gibi bileşenler içerirler.



Elektronik ve Dijital

- **Elektronik:** Bilgisayarlar, elektrik akımı kullanarak verileri işler.
- **Dijital:** Verileri sıfırlar (0) ve birler (1) olarak temsil eder.





Algoritma

- Adım adım bir problemi çözme prosedürüdür.
- Bir görevi nasıl gerçekleştireceğimizi belirlememize yardımcı olur.
- Bilgisayar programlarının temelini oluşturur.
- Bir problemi mantıklı ve etkili bir şekilde çözmek için kullanılırlar.
- Programlama, algoritmaları kullanarak program oluşturma sürecidir.
- İyi programcı, sorunu çözmek için uygun algoritmayı seçip, uygulayabilir.



Sabit Programlı Bilgisayar

- *Fixed program computer*, belirli bir problemi çözmek için tasarlanmıştır.
- Belirli bir görevi yapmak için önceden tanımlanmış bir programı çalıştırır.
- Kökleri çok eskilere dayanır.
- **Örneğin:** Abaküs, Antikythera Mekanizması, Pascaline, Leibniz Tekerleği, Jacquard'ın Dokuma Tezgahı, Babbage Fark Motoru, Hollerith Elektrik Çizelgeleme Sistemi, Atanasoff-Berry Bilgisayarı (ABC), Turing Bombası, ...





Charles Babbage ve Ada Lovelace

- Charles Babbage, ilk programlanabilir bilgisayar olan Analitik Makine'yi tasarlamıştır. Ada Lovelace, Analitik Makine için ilk programı yazmıştır.
- Sabit programlı bilgisayarlar,
 - Bilgisayar biliminin temellerini atmışlardır.
 - Teknolojik gelişmelere ilham kaynağı,
 - Modern bilgisayarların öncüleri olmuşlardır.
- Gelişmiş programlanabilir bilgisayarlar hala hayatımızın vazgeçilmez bir parçasıdır.



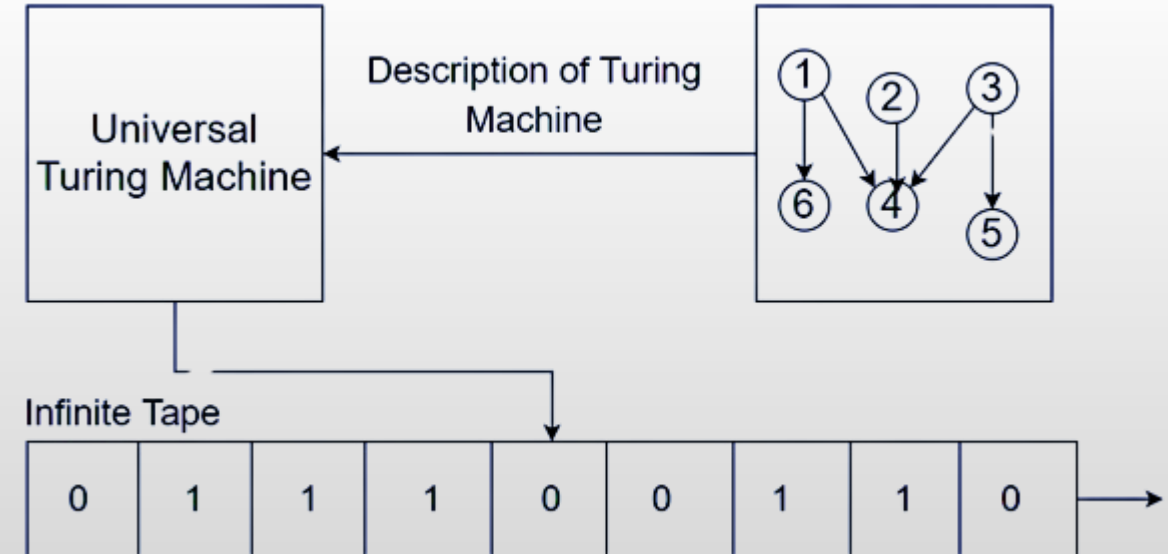
Depolanan Programlı Bilgisayar

- *Stored program computer*, belirli bir problemi çözmek için sabit bir program kullanmak yerine programlarını belleğinde depolayabilen makinelerdir.
- Daha esnek ve genel amaçlı bilgisayarların temelini atmıştır.
- Problemi çözmek için farklı programları çalıştırabilirler.
- Bilgisayarların daha önce çözemediği problemleri ele alabilmesini sağlar.
- Girdi, başka bir makine veya bir makinenin tanımı olabilir.
- *Evrensel Turing makineleri*, bu tür *makine girdi*'sini işleyebilir.



Evrensel Turing Makineleri

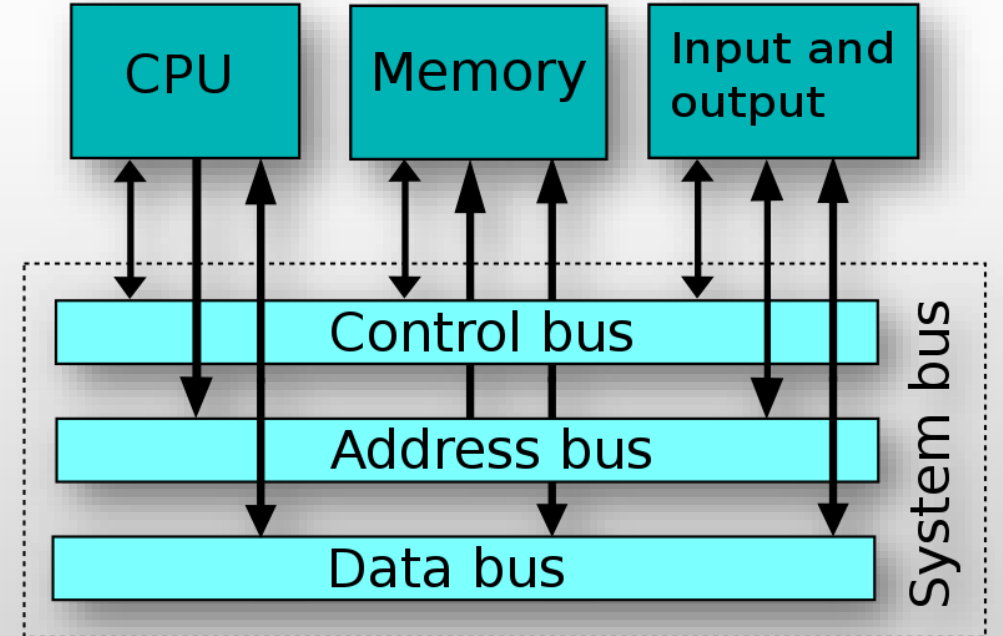
- Soyut ve genel amaçlı bilgisayardır.
- Bu kavram *Alan Turing* tarafından geliştirilmiştir.
- Herhangi bir Turing makinesini simüle edebilir, her tür problemi çözebilir.
- Modern bilgisayarların evrensel olarak programlanmasını sağlamıştır.
- Depolanan programlı bilgisayarlar ve Evrensel Turing makineleri, bilgisayar teknolojisinin gelişiminde önemli birer dönüm noktasıdır.





Genel Amaçlı Bilgisayar

- Kullanıcıların ihtiyaçlarına göre programlanabilir.
- Geniş bir uygulama yelpazesi için tasarlanmıştır.
- İş, eğitim, eğlence gibi birçok alanda kullanılabilirler.
- Programlar değiştirilerek,
 - farklı görevler için uyarlanabilirler.





Bilgisayarın İşlemesi

- Genel amaçlı bilgisayar, verileri işlemek için işlemci (CPU), bellek ve giriş/çıkış cihazları içerir.
- İşlemci, verileri talimatlara göre işler ve sonuç üretir.
- İş süreçlerini otomatikleştirir,
- Bilimsel hesaplamaları hızlandırır,
- İletişimi kolaylaştırır.



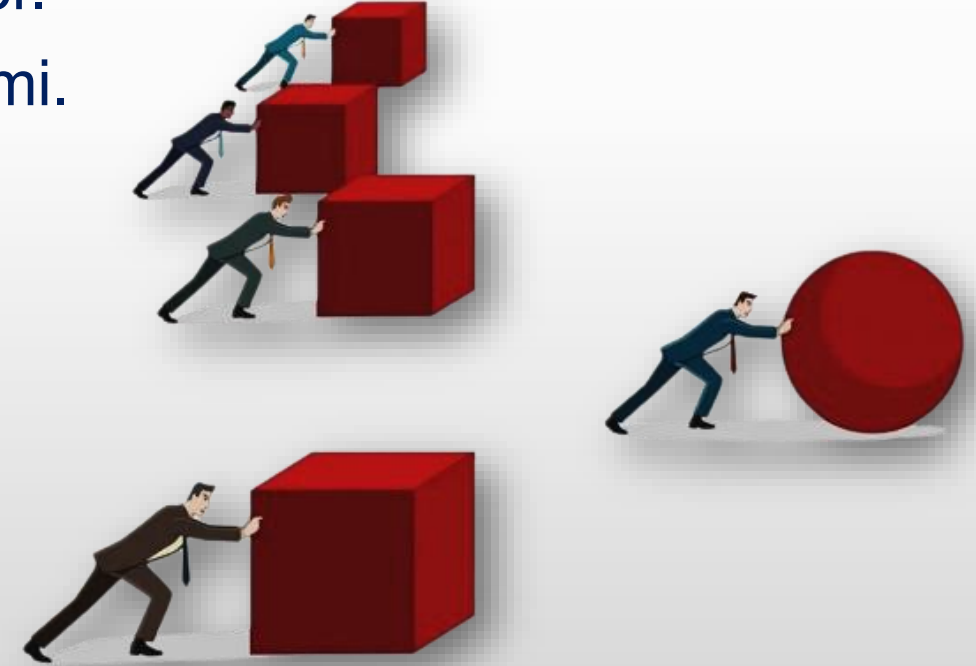
Bilgisayarların Sınırları

- Bilgisayarlar, karmaşık hesaplamaları hızlı bir şekilde gerçekleştirebilirler.
- Bu güç, teknolojik gelişmelere dayalıdır.
- İşlemci hızı, hesaplamaların ne kadar hızlı yapılacağını belirler.
 - Ancak işlemci hızı **sınırlıdır**.
- Bellek, bilgisayarın verileri saklama kapasitesini temsil eder.
 - Daha fazla bellek, daha karmaşık görevler demek.
 - Ancak, bellek kapasitesi **sınırlıdır**.



Bilgisayarların Sınırları

- Bilgisayarların veri işleme kapasitesi sınırlıdır.
- Büyük veri kümelerini işlemek için zaman ve kaynak gerekir.
- Bilgisayarlar, sınırlı kaynak olan enerji tüketirler.
- Daha güçlü bilgisayar, daha fazla enerji tüketimi.
- Programcılar, kaynakları etkin kullanarak bu sınırları aşmaya çalışırlar.





Makineler ve Teknoloji

- Teknoloji, insanların **yaşamını etkileyen** bir güçtür.
- Makineler, teknolojik gelişmenin önemli bir parçasını oluşturur.
- Makineler, işleri **hızlı ve verimli** bir şekilde yapabilirler.
- Bu **güç**, endüstriyel devrimden günümüze kadar sürekli artmıştır.
- Her şeye rağmen, inşa edebileceğimiz **makinelere 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 **fiziksel yasalara** bağlıdır.



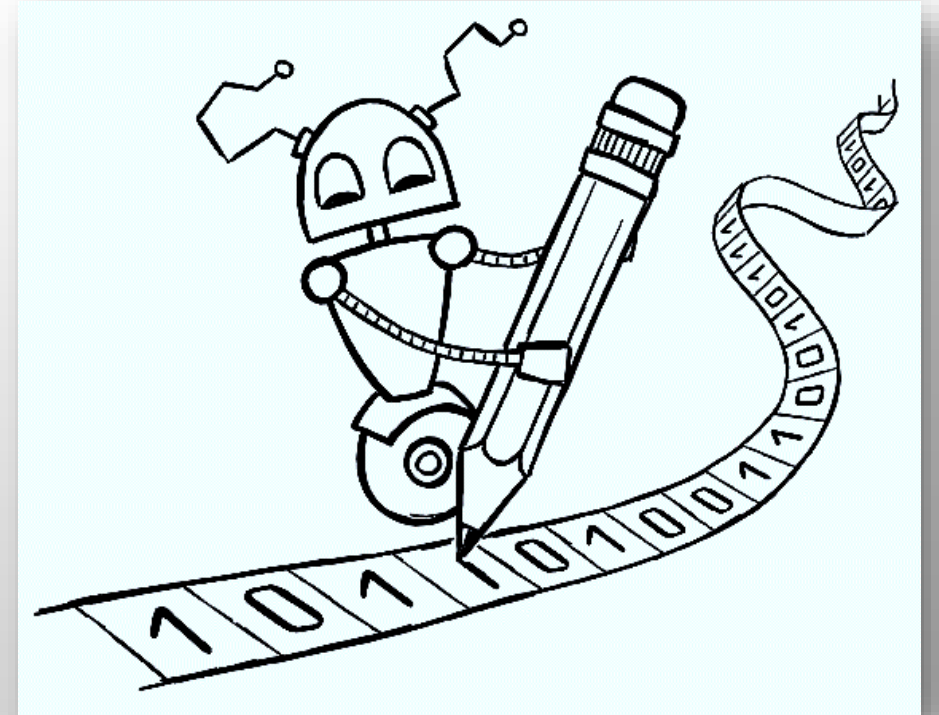
Makineler ve Teknoloji

- 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**.
- **Sınırları anlamak**, daha **iyi ve verimli** makineler tasarlamayı sağlayabilir.



Evrensel Turing Makineleri

- Bilgisayar biliminin en temel soyut modellerinden biridir.
- Herhangi bir *Turing* makinesinin işlemlerini simüle edebilir.
- **Bant (Tape)**
 - İşlem için bir bant kullanılır.
 - Bant, girdi, çıktı ve ara sonuçları saklar.
 - Bant, birbirine bağlı hücrelerden oluşur.
 - Her bir hücreye bir sembol yazılabilir.





Evrensel Turing Makineleri

- **Bant Başlığı (Tape Head)**
 - Bant üzerindeki bir hücreyi işaret eder.
 - Aktif hücredeki sembolü okur, yerine yeni bir sembol yazar.
 - Bir hücre ileri veya geri hareket eder.
- **Hücreler ve Semboller**
 - Bant hücreleri, işlemi yönlendirmek için birer sembol içerir.
 - Bir sembol, mevcut durumu ve hangi işlemin yapılacağını belirler.



Evrensel Turing Makineleri

- Hesaplamaların sınırlarını tanımlar.
- Bu model, hesaplanabilir her şeyi hesaplayabilir.
- 20. yüzyılın en önemli bilimsel sonuçlarından biri olarak kabul edilir.



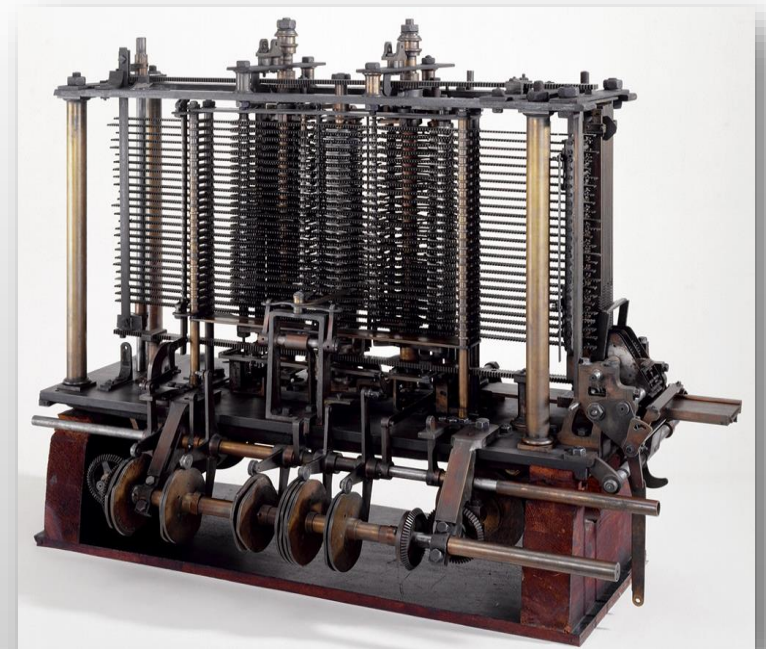
Ada Lovelace Kimdir?

- 19. yüzyılın ilk yarısında yaşamış İngiliz matematikçi ve yazardır.
- Analitik Makine için algoritma ve program yazmıştır.
- Bilgisayar tarihindeki ilk programcılardan biridir.
- Matematiksel düşünce ve programlamayı bilimsel yaklaşımla birleştirmiştir.



Babbage'ın Analitik Makinesi

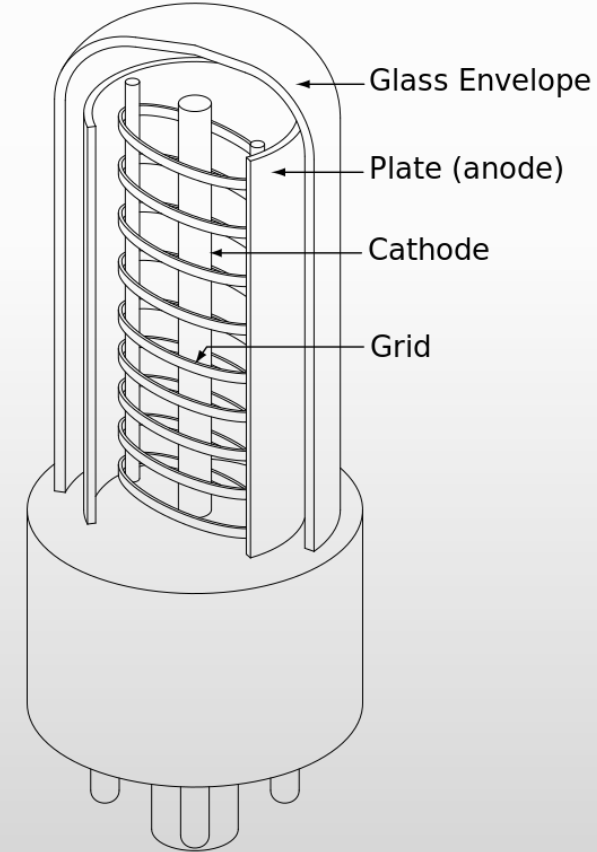
- Charles Babbage 19. yüzyılda yaşamış İngiliz matematikçi ve mucit.
- Analitik Makine'yi tasarlamış ve geliştirmiştir.
- Matematiksel hesaplama yapabilen ve sonuçları saklayabilen bir cihazdı.
- Farklı hesaplamaları yapmak için programlanabilirdi.
- Modern bilgisayarların öncüsüdür.





Colossus Mark 1

- 1944 yılında inşa edilen programlanabilir ilk elektronik dijital bilgisayar.
- Düşmanın iletişimini bozmak için gerekli gizli kodları çözdü.
- Elektronik valfler (vakum tüpleri) kullanıyordu.
- Mekanik cihazlara göre daha hızlı ve esnek.
- Sadece savaş sırasında değil, bilgisayar teknolojisinin ilerlemesine de büyük katkıda bulundu.





ENIAC

- 1946'da ABD'de *John Mauchly* ve *J. Presper Eckert* tarafından geliştirildi.
- Genel amaçlı bir elektronik bilgisayardı.
- Farklı işlemleri gerçekleştirebilecek şekilde tasarlanmıştı.
- Mekanik parçalar yerine elektronik valfler (*vakum tüpleri*) kullanıyordu.
- Bu, daha hızlı ve daha güvenilir bir işlem yapma yeteneği sağladı.
- Büyük ölçekli ve hızlı hesaplamalar için kullanılabilirdi.
- İlk büyük ölçekli elektronik bilgisayar olarak kabul edilir.



EDVAC

- 1951 yılında ABD'de *John von Neumann* tarafından geliştirildi.
- ENIAC'tan farklı olarak, ikili sayı sistemi kullanıyordu.
- İkili sistem, yalnızca 0 ve 1'leri kullanarak bilgiyi temsil eder.
- Programları ve verileri bellekte sıralı olarak saklıyordu.
- Bu, daha karmaşık ve esnek programların yazılabilmesini sağladı.
- Komutlar, sıralı olarak çalışırdı, koşula dayalı atlama yapabiliyordu.
- Von Neumann mimarisi, modern bilgisayarların temelini atmıştır.



Bilgi

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



Bildirimsel Bilgi

- Açıkça ifade edilen bilgi türüdür.
- Doğru veya yanlış olarak doğrulanabilir.
- Aksiyomlar veya tanımlar yoluyla ifade edilebilir.
 - Aksiyomlar, bir şeyin ne olduğunu veya nasıl tanımlandığını belirtir.
- Aynı zamanda gerçeklerin ifadesini içerebilir.
 - Gerçekler, dünya hakkında doğru ve doğrulanabilir ifadelerdir.
- **Örnek Bir Aksiyom**
 - " y , x 'in kareköküdür, $y*y = x$ ise" açık bir tanımlamayı ifade eder.
 - Ancak, karekök hesaplamasının nasıl yapılacağı hakkında bilgi vermez.



Buyurusal Bilgi

- *Bir şeyin nasıl yapılacağı* konusunda talimatlar içerir.
- *Babylonian* yöntemi ile bir sayının karekökü hesaplanabilir.
- Yöntemin Adımları
 1. x değerini al.
 2. bir başlangıç değeri olan y_0 ile başla.
 3. Eğer $y_n^2 \approx x$ ise, işlem sona erer.
 4. Değilse, $y_{n+1} = (y_n + x/y_n)/2$ ile yeni bir değer hesapla.
 5. Adım (3)'ü tekrarla.
- Başlangıç tahminiyle başlar. Karekökün yaklaşık değeri hesaplanır. Adımlar, 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 bölünebilen en büyük pozitif tamsayıdır.
- **Bildirimsel Tanım**
 - GCD'nin ne olduğunu açıklar.
 - "d, a ve b'nin GCD'si ise, d değeri $a = dx$ ve $b = dy$ eşitliklerini sağlar."
- **Buyurusal Tanım: Öklidyen Algoritması**
 - İki sayının GCD'sini bulmak için kullanılan bir algoritmadır.
 - Nasıl yapılacağına dair talimatlar içerir.



Öklidyen Algoritması

▪ Algoritmanın Adımları

1. İki pozitif tam sayı olan a ve b 'yi al ($a \geq b$).
2. a 'yı b 'ye böl ve kalanı R olarak adlandır.
3. Eğer $R = 0$ ise, işlem sona erer.
4. Değilse, a 'yı b 'ye, b 'yi de R 'ye eşitle.
5. Adım (3)'ü tekrarla.



Bilgisayar Biliminde Neler Var?

- Bilgisayar bilimi bir bilim dalıdır.
- Gerçekliği anlama, modelleme ve analiz etme amacı güder.
- Soyutlama, problem çözme, ve yaratıcılığı temsil eder.
- Soyutlama, karmaşık bir sistemi veya problemi basitleştirmek için kullanılır.
- Problemleri çözmek için algoritmalar ve kod kullanılır.
- Mantık ve analitik düşünceyle sınırlı değildir.
- Sanat, yaratıcılık ve tasarım da bu alanda büyük bir rol oynar.
 - Örneğin, dijital medya, elektronik müzik, oyunlar, animasyon..



Algoritmalar

- Bir sorunu çözmek için izlenmesi gereken adımları belirleyen bir tariftir.
- Karmaşık sorunları basitleştirmek ve çözmek için kullanılır.
- Bilgisayar bilimi, temelde algoritmaların çalışma ve analizini içerir.
- Algoritmalar,
 - Adımların belirlenmesi,
 - Mantıklı bir sıra oluşturulması ve
 - Problemi çözme amacı güder.
- İyi tasarlanmış algoritmalar, verimlilik ve doğruluk açısından önemlidir.



Problem Belirleme

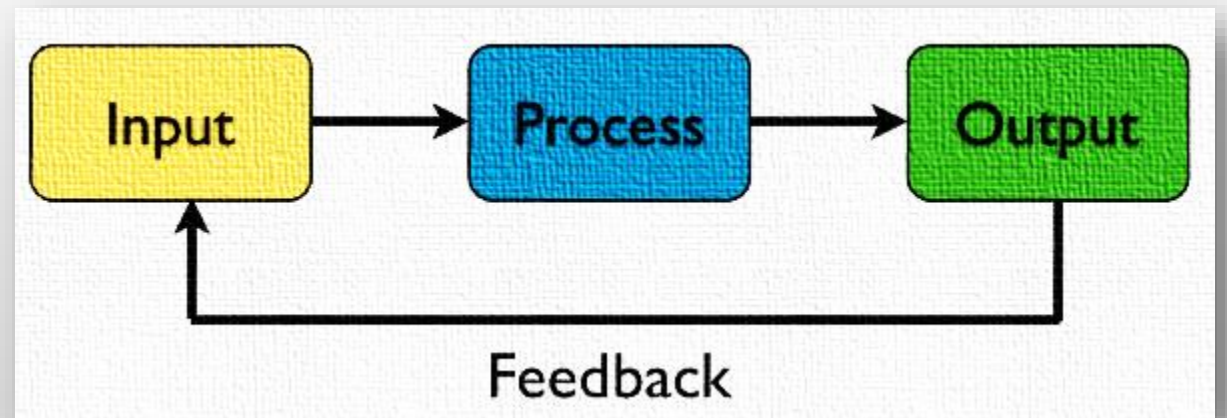
- Bir işin başarılı olabilmesi için, sorunun ne olduğu açıkça belirlenmelidir.
- Problem belirleme, bir sorunun doğru bir şekilde tanımlanmasını sağlar.
- Algoritmalar, bu belirlenen problemleri çözmek için kullanılır.
- Problem belirleme,
 - Problemin boyutunun ve
 - Karmaşıklığının anlaşılmasına yardımcı olur.





Problem Belirleme

- Bir programın veya algoritmanın ne yapması gerektiği açıkça tanımlanır.
- Gerekli olan adımlar ve çıktı belirlenir.
- Problem belirlerken, iki bileşen vardır: *girdi (input)* ve *çıktı (output)*.
 - **Girdi**, problemin çözülmesi için gereken bilgi veya verileri temsil eder.
 - **Çıktı**, problemin sonucunu veya istenen bilgiyi temsil eder.





Örnek: İki Sayının Toplamı

- Örnek problem açıklaması:
 - **Problem:** Verilen iki sayının toplamını bulma.
 - **Girdi:** İki sayı, X ve Y .
 - **Çıktı:** İki sayının toplamı olan bir sayı, Z .
- Eğer $X = 5$ ve $Y = 3$ ise, $Z = 8$ olmalıdır.



Sözde Kod (Pseudocode)

- Algoritmalar, kolay anlaşılabilmesi için sözde kod kullanılarak ifade edilir.
- Sözde kod,
 - İnsan diline benzer bir dil kullanır ve
 - Algoritmanın mantığını açıklar.



Sözde Kod

- **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.



Sözde Kod

- Programlama dili yerine konuşma dili kullanılır.
- Adımlar, genellikle numaralandırılmış ve girintili olarak sunulur.
- Çoğu işlem için sabit bir sözdizimi gerekmez.
- Doğal dile kıyasla daha açık ve okunaklıdır.
- İşlemi daha anlaşılır hale getirir.
- Mantık hatalarını tespit etmeyi kolaylaştırır.
- Programcılar arasında işbirliğini kolaylaştırır.
- Algoritma davranışı hakkında mantıklı çıkarımlara izin verir.
- Kod yazma aşamasına geçişi kolaylaştırır.



Algoritmik İşlem Türleri

- Algoritmalar, **sıralı**, **koşullu** ve **döngüsel** işlemleri kullanır.
- Bu işlem türleri, her programlama dilinde bulunur.
- Temel programlama becerilerinin bir parçasıdır.
- Algoritmaların temel yapısını oluşturur.
- Karmaşıklığı azaltmak ve işlemleri kontrol etmek için kullanılır.



Algoritmik İşlem Türleri

- **Sıralı İşlemler (*Sequential Operations*)**
 - Her işlem bir öncekinden sonra gelir ve sırayla çalışır.
 - **Örnek:** Verilerin okunması, işlenmesi ve sonuçların üretilmesi.
- **Koşullu İşlemler (*Conditional Operations*)**
 - Eğer 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*)**
 - Belirli bir koşul karşılanana kadar belirli bir işlemi tekrarlar.
 - **Örnek:** Bir dizi elemanın tümünün toplanması.



Sıralı İşlemler: Giriş, Hesaplama ve Çıkış

- Programın adım adım sırayla gerçekleştirilen işlemlerini ifade eder.
- **Giriş İşlemleri**
 - Dış dünyadan veri almak için kullanılır.
 - **Örnek:** Bir kişinin ağırlığını (w) ve boyunu (h) almak.
- **Hesaplama İşlemleri**
 - Değişkenlerin değerlerini atamak ve aritmetik işlemler için kullanılır.
 - **Örnek:** BMI (Vücut Kitle İndeksi) hesaplamak için $h / (w * w)$ işlemi.
- **Çıkış İşlemleri**
 - Sonuçları dış dünyaya göndermek ve görüntülemek için kullanılır.
 - **Örnek:** BMI'nin (vücut kitle indeksi) değerini yazdırmak.



Örnek - İki Sayıyı Toplama

- Adım 1: başla
- Adım 2: birinci sayıyı al (a)
- Adım 3: ikinci sayıyı al (b)
- Adım 4: c 'yi hesapla ($c = a + b$)
- Adım 5: c 'yi yazdır
- Adım 6: bitir



Örnek - Dikdörtgenin Alanı

- Adım 1: başla
- Adım 2: dikdörtgenin tabanını al (b)
- Adım 3: dikdörtgenin yüksekliğini al (h)
- Adım 4: alanı hesapla ($\text{alan} = b * h$)
- Adım 5: alanı yazdır
- Adım 6: bitir



Koşullu İşlemler: Soru Sor ve Alternatif Eylemler Seç

- Programın soru sorup, cevaplara göre farklı eylemlerde bulunması.
- *Eğer* şart sağlanıyorsa şunu yap, aksi takdirde başka bir şey yap.
 - Eğer x 100'den büyükse, x 'i yazdır; aksi takdirde x 'e 100 ekle.
- Karmaşık soruları destekler.
 - Eğer x 100'den büyükse ve y 200'e eşitse, o zaman...
- Cevaplar *Doğru* veya *Yanlış* olur.
- Programın durumlar arasında geçiş yapmasını sağlar.



Örnek - İki Sayıyı Al, Daha Büyüğünü Yazdır

- Adım 1: başla
- Adım 2: ilk sayıyı al (f)
- Adım 3: ikinci sayıyı al (s)
- Adım 4: ilk sayı daha büyükse ilk sayıyı yazdır
- Adım 5: aksi takdirde ikinci sayıyı yazdır
- Adım 6: bitir



Döngülü İşlemler: Tekrar Eden Eylemler

- Devam koşulu yanlış olana kadar belirli eylemleri tekrar eder.
- Bir işlemi tekrar etmek veya döngü içinde çalışmak için kullanılır.
- Verileri işlemek, listeleri gezmek gibi işlemlerde kullanışlıdır.



Döngülü İşlemler: Tekrar Eden Eylemler

- Eğer i değeri 0'dan büyükse, s 'yi 1 arttır ve i 'yi 1 azalt.
- Bu işlemi, i değeri 0'dan büyük olduğu sürece tekrarla.

while $i > 0$ **do**

set s **to** $s + 1$

set i **to** $i - 1$

- Bir işlemi k değeri n değerinden büyük olana kadar tekrarla.

repeat

print k

set k **to** $k + 1$

until $k > n$



Örnek - Bir Listede En Büyük Öğeyi Bulma

- Adım 1: başla
- Adım 2: *en büyük değer* olarak 0 ata
- Adım 3: kontrol edilecek öğe varsa, devam et.
- Adım 3: öğe daha büyükse *en büyük değer*'i güncelle.
- Adım 4: her döngüde *en büyük değer*'i yazdır.
- Adım 5: bitir



Koşullu ve Döngülü İşlemler

- Koşullu işlemler, belirli koşullara göre eylemleri kontrol eder.
- Döngülü işlemler, belirli koşullara bağlı olarak eylemleri tekrarlar.
- Programların daha esnek ve güçlü olmasını sağlar.
- Doğru koşullar ve iyi tasarlanmış döngüler, beklenen çıktıyı verir.
- Sonsuz döngüler hatalara neden olabilir.





Koşullu ve Döngülü İşlemler

- Bir döngünün iki temel bileşeni vardır:
 - **Koşul (*Continuation Condition*)**: Ne zaman sona ereceğini belirler.
 - **Gövde (*Loop Body*)**: Her tekrarda gerçekleştirilen eylemleri içerir.
- **Sonsuz Döngü**
 - Devam koşulu hiçbir zaman yanlış (*false*) olmayan döngü türüdür.
- **Sonsuz Döngü Hatası**
 - Programı çökertecek veya yanıt vermeyen bir duruma getirebilir.



SON