

BLM2005 Nesne Yönelimli Programlama

Dr. Öğr. Üyesi Eyüp Emre ÜLKÜ

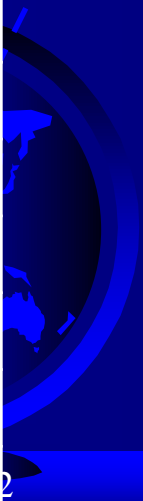
Derslik: T4-Z06

Ders Saati:2+2

Kredi:6 ECTS:6

Dersin İçeriği

1	Dersin Tanıtımı
2	Java Programlamaya Giriş
3	Nesneye Dayalı Programlaya Giriş
4	Sınıf ve Nesne Kavramları
5	Kalıtım ve Çok Biçimlilik
6	Kalıtım ve Çok Biçimlilik
7	İstisna Yönetimi ve Metin I/O
8	Ara Sınav
9	Soyut Sınıflar ve Arayüzler
10	Olaya-Dayalı Programlama
11	Kullanıcı Arayüzleri Oluşturma
12	Özyineli (Recursive) Fonksiyonlar
13	Generic Sınıflar
14	Tasarım Şablonlarına Giriş
15	Konu Tekrarı



Ders Kaynakları

- Ders sunumları
- Y. Daniel Liang , Introduction to Java Programming, Comprehensive, 8/E, Prentice Hall, 2011.
- Herbert Schildt, Herkes için Java, Alfa Yayınları, 2007
- Ali Yazıcı, Erdoğan Doğdu ve arkadaşları, Java Bilgisayar Programlamaya Giriş, Palme yayıncılık, 2008.



Değerlendirme

Türü	Sayı	Katkısı (%)
Ara sınav	1	25
Lab Quizler	3	10
Lab Proje	1	15
Final Sınavı	1	50

NOT: Katılım zorunluluğu %70

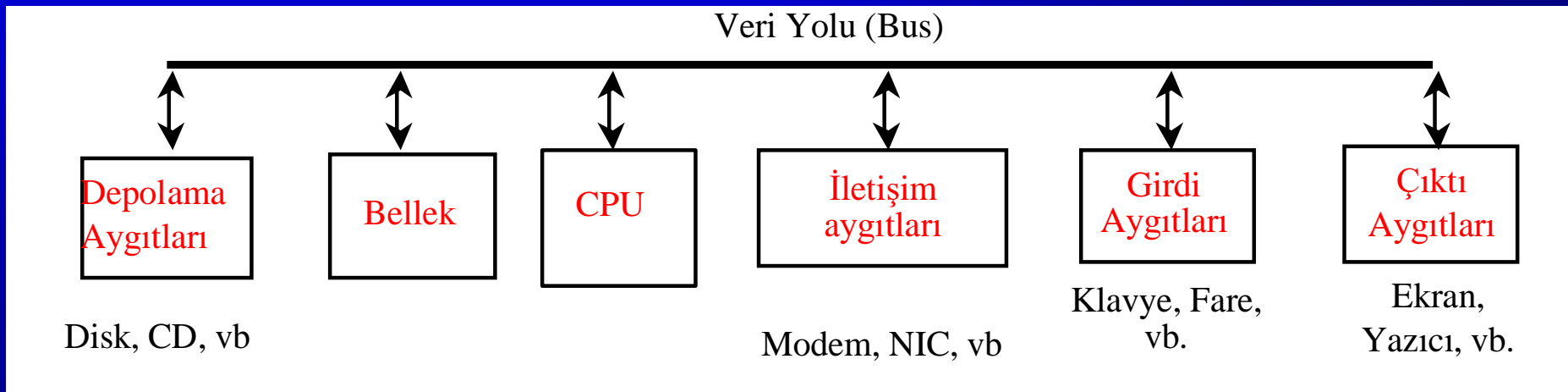


Bölüm 1: Bilgisayara ve JAVA'ya Giriş



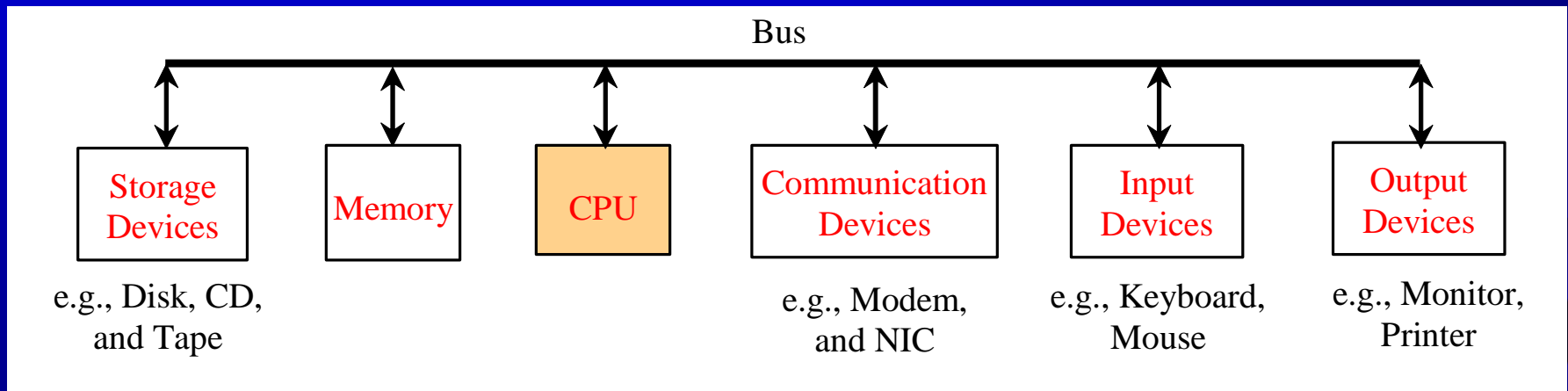
Bilgisayar Nedir?

Bilgisayar, işlemci (CPU), hafıza, sabit disk, ekran, yazıcı ve iletişim aygıtlarının oluşturduğu bir bütündür.



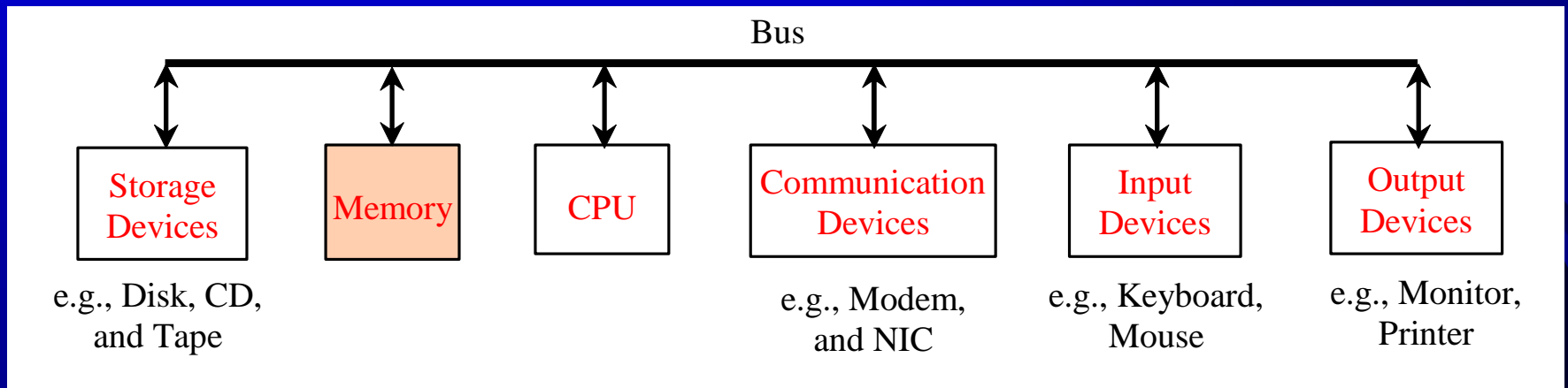
İşlemci (CPU)

Bilgisayarın beyni olarak bilinir. Bellekten komutları alır ve çalıştırır.



Bellek

Bellek, işlemcinin çalıştırması için veri ve program komutlarını tutar. Bellek birimi sıralı bir byte dizisidir ve her bir byte 8 bit'tir . Bir program çalıştırılmadan önce program ve verisi belleğe aktarılır. Bellekte her zaman bir şey tutulur. Ancak bunun her zaman bir anlamı olmayabilir. Yeni bir bilgi geldiğinde bellekteki eski bilgi silinir.



Bellekte Veri Nasıl Tutulur?

Veri; sayı, karakter, dizge(string) farklı türlerde olabilir. Bu veriler, 0 ve 1'lerden oluşan bit dizileri olarak kodlanır.

Bütün dijital aygıtlar için 2 kararlı durum söz konusudur. Bu nedenle bilgisayarlar 0 ve 1 kullanır.

Kodlama şemaları sistem tarafından otomatik olarak oluşturulur ve değişebilir.

Memory address	Memory content	
.	.	
.	.	
.	.	
2000	01001010	Encoding for character 'J'
2001	01100001	Encoding for character 'a'
2002	01110110	Encoding for character 'v'
2003	01100001	Encoding for character 'a'
2004	00000011	Encoding for number 3



Bellekte Veri Nasıl Tutulur?

Örneğin:

‘J’ karakteri 1 byte içerisinde 01001010 olarak kodlanır.

3 sayısı 00000011 olarak çok basit biçimde kodlanır. (2lik taban)

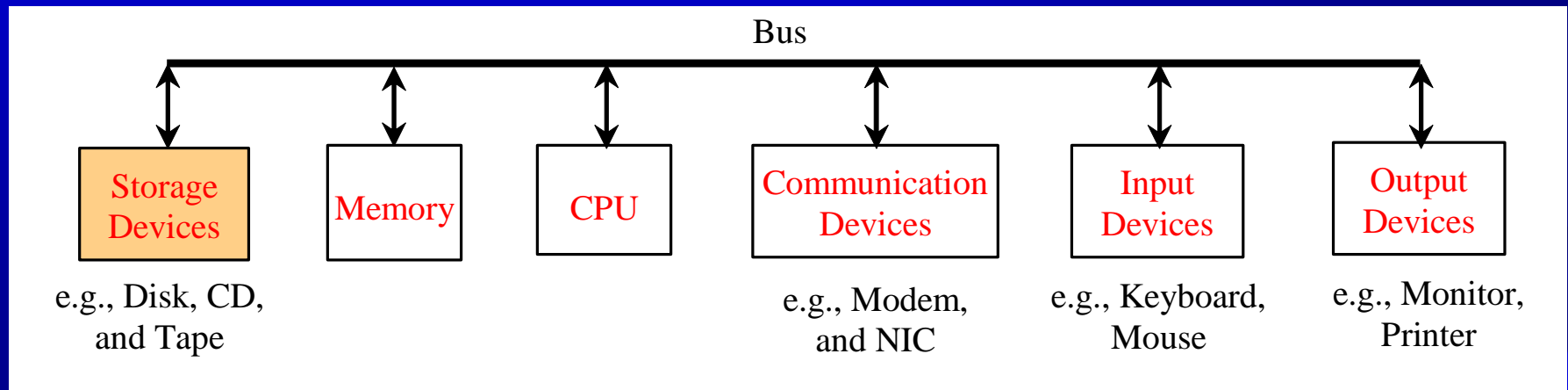
Byte en küçük saklama birimidir. Eğer bilgisayar 1 byte’den daha büyük veri tutmak isterse bitişik byte dizisi kullanır. 2 veri aynı byte içinde tutulmaz.

Memory address	Memory content	
.	.	
.	.	
.	.	
2000	01001010	Encoding for character ‘J’
2001	01100001	Encoding for character ‘a’
2002	01110110	Encoding for character ‘v’
2003	01100001	Encoding for character ‘a’
2004	00000011	Encoding for number 3



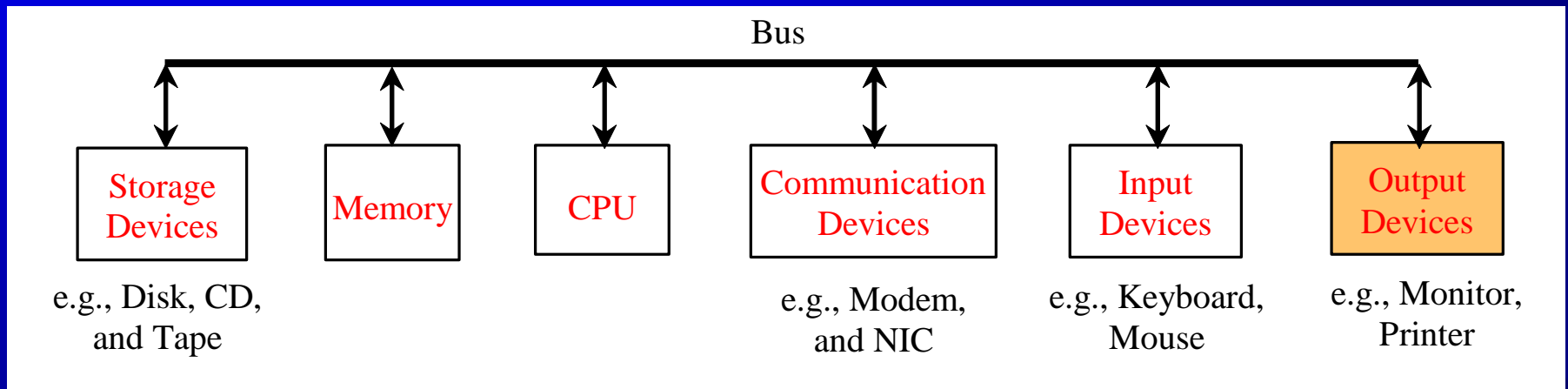
Saklama/Depolama aygıtları

Bellekde tutulan bilgi geçicidir çünkü güç kaynağı kesildiğinde bellekteki bilgi de gider. Programlar ve veriler kalıcı olarak depolama aygıtlarında tutulur ve kullanılmak istenildiğinde belleğe alınır. Temelde 3 tip saklama aygıtı bulunmaktadır. Bunlar: disk sürücüler (hard disks and floppy disks), CD sürücüler (CD-R and CD-RW), ve Teyp sürücüler.



Çıktı Aygıtları

Çıktı aygıtları ile bilgi metin veya grafik olarak gösterilir.
Çözünürlük ve nokta uzaklığı görüntünün kalitesini belirlemektedir.



Programlar

Bilgisayar *programları* diğer adıyla *yazılımlar* bilgisayara ne yapması gerektiğini söyleyen kodlanmış komut dizisidir.

Yazılım olmadan bilgisayarın hiç bir işlevi yoktur.
Bilgisayarın iletişim dili yazılımlardır.

Programlar/Yazılımlar ise programlama dili kullanılarak yazılır.



Programlama Dilleri

Makine Dili

Assembly Dili

Yüksek-Seviyeli Diller

Makine dili, bilgisayarlar içerisine kurulmuş ilkel komutlar dizisidir. Bu komutlar ikilik tabanda yazılmış, 0 ve 1 den oluşan kodlardır. Makine dili ile program yazmak çok zordur. Ayrıca makine dili ile yazılmış programların okunması ve değiştirilmesi çok zordur. Örneğin iki sayıyı toplamak için aşağıdaki gibi bir kod yazmanız gerekir.

```
1101101010011010
```



Programlama Dilleri

Makine Dili

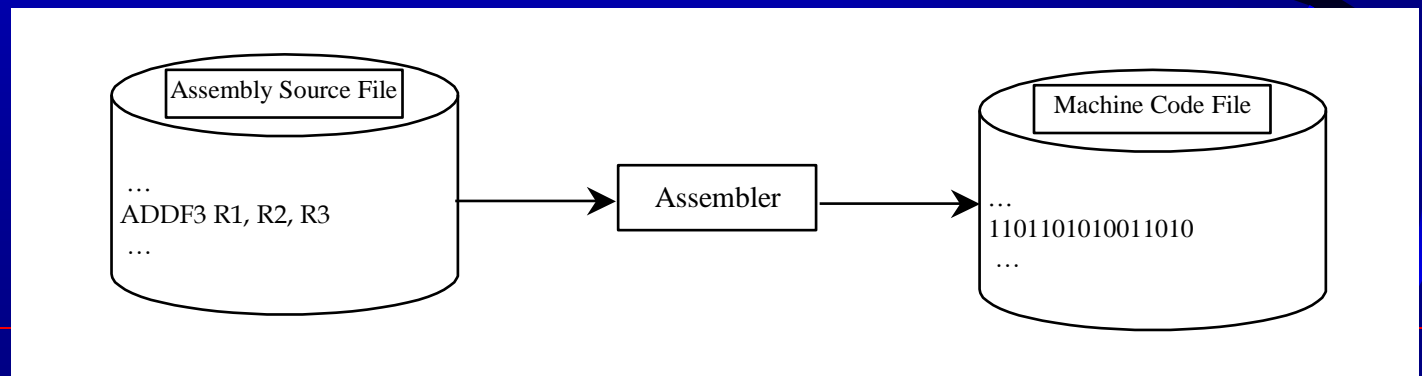
Assembly Dili

Yüksek-Seviyeli Diller

Assembly dilleri programlamayı kolaylaştırmak amacı ile geliştirilmiştir. Bilgisayar Assembly dilini anlamak için, assembly dilindeki programları makinanın anlayacağı şekilde dönüştüren assembler adında bir program kullanır.

Örneğin, assembly dilinde iki sayıyı toplamak :

`ADDF3 R1, R2, R3`



Programlama Dilleri

Makine Dili

Assembly Dili

Yüksek-Seviyeli Diller

Yüksek-seviye programlama dilleri öğrenmesi ve kolay dillerdir.

Örneğin yüksek seviyeli bir dilde yarıçapı 5 olan bir daire alanını hesaplamak aşağıdaki gibidir

$\text{Alan} = 5 * 5 * 3.1415;$



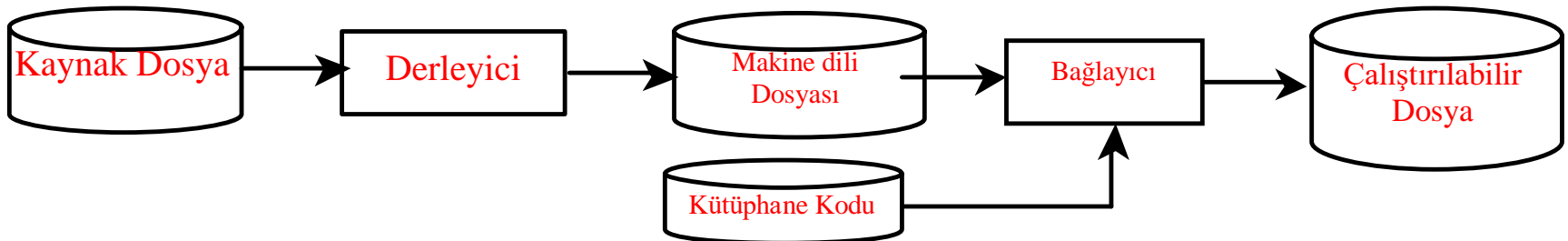
Yüksek-Seviyeli Programlama Dilleri

- COBOL (COmmon Business Oriented Language)
- FORTRAN (FORmula TRANslation)
- BASIC (Beginner All-purpose Symbolic Instructional Code)
- Pascal (named for Blaise Pascal)
- Python (Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.)
- C (whose developer designed B first)
- Visual Basic (Basic-like visual language developed by Microsoft)
- Delphi (Pascal-like visual language developed by Borland)
- C++ (an object-oriented language, based on C)
- C# (a Java-like language developed by Microsoft)
- Java (We use it in the book)



Kaynak Kod Derleme

Yüksek-seviyeli programlama dili kullanılarak yazılan bir program kaynak kod olarak adlandırılır. Bilgisayarın kaynak kodu anlaması için makine diline çevirilmesi gerekmektedir. Bunu sağlayan ise derleyicidir. Derleyici bir nesne kodu(*object program*) üretir. Nesne kodu, ana kodun çalıştırılması için gerekli kütüphane kodları ile bağlantı sağlayarak kodu çalıştırılabilir (executable) dosya haline getirir.

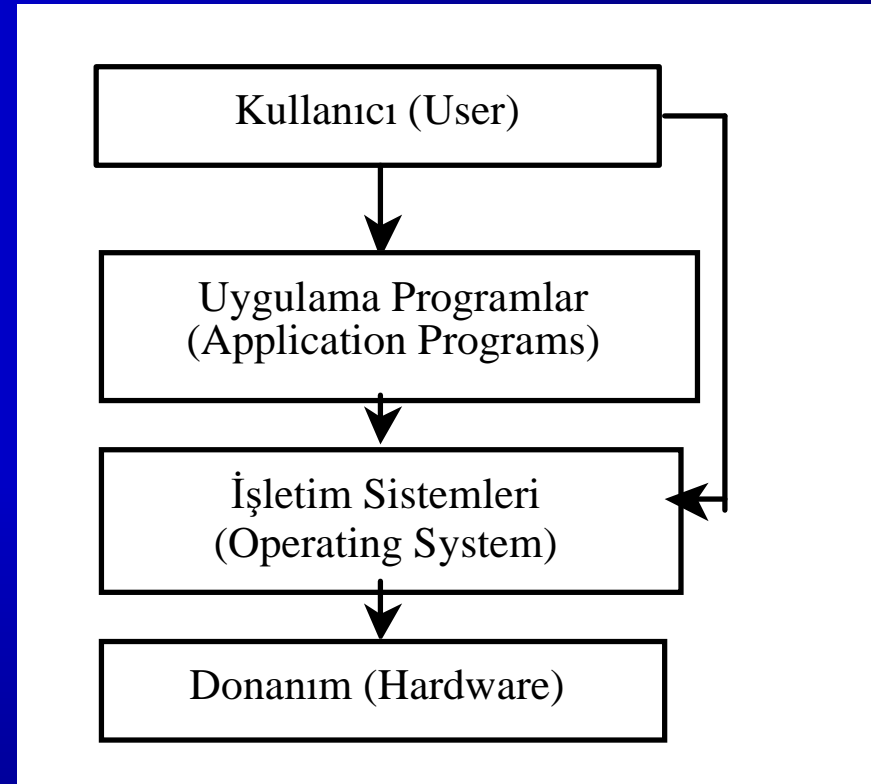


İşletim Sistemleri (Operating Systems)

İşletim sistemi bilgisayarı kontrol eden ve yöneten bir programdır.

Günümüzde en yaygın kullanılan işletim sistemi Window'tur.

İşletim sistemi olmadan uygulama programları (İnternet Tarayıcı, Word vb.) çalıştırılmaz



Neden Java?

Java, kullanıcılara internet üzerinden sunucular, masaüstü bilgisayarlar, elle tutulan cihazlar için uygulama geliştirimi ve dağıtımını sağlar. Bilgisayarın geleceği internetten etkilenmektedir. Bu nedenle Java geleceğin bir parçası olarak kalmayı vaad etmektedir.

- Java genel amaçlı bir programlama dilidir.
- Java internet programlama dilidir. .



Java, Web

- Java uygulamaları
- Java web uygulamaları ve geliştirme
- Java ile cep telefonları ve elle tutulabilir cihazlar için uygulama geliştirilebilir.



PDA, Cep telefonu ve Akıllı telefon



Java'nın Özellikleri

- ❑ Java basit bir dildir
- ❑ Java nesne yönelimlidir
- ❑ Java dağıtıktır
- ❑ Java yorumlanabilirdir
- ❑ Java güçlü bir programlama dilidir
- ❑ Java güvenli bir programlama dilidir
- ❑ Java platformdan bağımsızdır
- ❑ Java taşınabilirdir.
- ❑ Java işlevseldir.
- ❑ Java çoklu kullanım özelliği
- ❑ Java dinamiktir



www.cs.armstrong.edu/liang/intro8e/JavaCharacteristics.pdf

Java'nın Özellikleri

- ❑ **Java basit bir dildir**
 - ❑ Java nesne yönelimlidir
 - ❑ Java dağıtıktır
 - ❑ Java yorumlanabilirdir
 - ❑ Java güçlü bir programlama dilidir
 - ❑ Java güvenli bir programlama dilidir
 - ❑ Java platformdan bağımsızdır
 - ❑ Java taşınabilirdir.
 - ❑ Java işlevseldir.
 - ❑ Java çoklu kullanım özelliği
 - ❑ Java dinamiktir
- Java kısmen C++ model alınarak ortaya çıkarılmıştır ancak daha geliştirilmiş ve basitleştirilmiştir. Bazı programcılar tarafından Java, C++'a benzerliği yanı sıra daha fazla işlevsel ve daha az olumsuz yönleri olması nedeniyle, "C++--" olarak adlandırır.



Java'nın Özellikleri

- ❑ Java basit bir dildir
- ❑ **Java nesne yönelimlidir**
- ❑ Java Java dağıtıktır
- ❑ Java yorumlanabilirdir
- ❑ Java güçlü bir programlama dilidir
- ❑ Java güvenli bir programlama dilidir
- ❑ Java platformdan bağımsızdır
- ❑ Java taşınabilirdir.
- ❑ Java işlevseldir.
- ❑ Java çoklu kullanım özelliği
- ❑ Java dinamiktir

Java doğası gereği nesne yönelimli bir programlama dilidir. Çoğu nesne yönelimli programlama dilleri yordamsal dil (procedural languages) olarak ortaya çıkmasına karşın, Java baştan beri nesne yönelimli programlama olarak dizayn edilmiştir.. Nesne yönelimli programlama (OOP), geleneksel yapısal dillerin yerini alan popüler bir programlamadır.

Yazılım geliştirmede en temel konulardan biri kodun nasıl yeniden kullanılacağıdır. Nesne yönelimli programlama, kapsülleme (encapsulation), kalıtım (inheritance) ve çokbiçimlilik (polymorphism) özellikleri ile çok büyük esneklik, birimsellik, anlaşılabilirlik ve tekrar kullanılabilirlik sağlamaktadır.

Java'nın Özellikleri

- ❑ Java basit bir dildir
- ❑ Java nesne yönelimlidir
- ❑ Java dağıtıktır.
- ❑ Java yorumlanabilirdir
- ❑ Java güçlü bir programlama dilidir
- ❑ Java güvenli bir programlama dilidir
- ❑ Java platformdan bağımsızdır
- ❑ Java taşınabilirdir.
- ❑ Java işlevseldir.
- ❑ Java çoklu kullanım özelliği
- ❑ Java dinamiktir

Dağıtık hesaplama bir network üzerinde birçok bilgisayar ile çalışmak demektir. Java dağıtık hesaplamayı kolaylaştıracak şekilde dizayn edilmiştir. Ağ iletişimi Java içerisine yapısı gereği entegre edilmiştir. Bu nedenle network programları bir dosyadan/dosyaya veri alıp gönderme gibidir.



Java'nın Özellikleri

- ❑ Java basit bir dildir
- ❑ Java nesne yönelimlidir
- ❑ Java dağıtıktır
- ❑ **Java yorumlanabilirdir.**
- ❑ Java güçlü bir programlama dilidir
- ❑ Java güvenli bir programlama dilidir
- ❑ Java platformdan bağımsızdır
- ❑ Java taşınabilirdir.
- ❑ Java işlevseldir.
- ❑ Java çoklu kullanım özelliği
- ❑ Java dinamiktir

Java programlarını çalıştırmak için bir yorumlayıcıya (interpreter) ihtiyacınız vardır. Programlar, bytecode adı verilen Java Sanal Makinası (Java Virtual Machine) kodu olarak derlenir. Bu kod makineden bağımsız olup, JVM'nin bir parçası olan Java yorumlayıcısının bulunduğu her makinada çalıştırılabilir.



Java'nın Özellikleri

- ❑ Java basit bir dildir
- ❑ Java nesne yönelimlidir
- ❑ Java dağıtıktır
- ❑ Java yorumlanabilirdir
- ❑ Java güçlü bir programlama dilidir.
- ❑ Java güvenli bir programlama dilidir
- ❑ Java platformdan bağımsızdır
- ❑ Java taşınabilirdir.
- ❑ Java işlevseldir.
- ❑ Java çoklu kullanım özelliği
- ❑ Java dinamiktir

Java derleyicileri, çalıştırıldıklarında bir çok problemi tespit edebilme özelliğine sahiptir.

Java, diğer programlama dillerinde bulunan hata eğilimli yapılardan arındırılmıştır.

Java olağandışı durumlarda (hata oluştuğunda) kural dışı çalışma özelliğine sahiptir. (exception-handling feature)



Java'nın Özellikleri

- ❑ Java basit bir dildir
- ❑ Java nesne yönelimlidir
- ❑ Java dağıtıktır
- ❑ Java yorumlanabilirdir
- ❑ Java güçlü bir programlama dilidir
- ❑ Java güvenli bir programlama dilidir.
- ❑ Java platformdan bağımsızdır
- ❑ Java taşınabilirdir.
- ❑ Java işlevseldir.
- ❑ Java çoklu kullanım özelliği
- ❑ Java dinamiktir

Java sisteminizi başıboş programların neden olduğu zararlara karşı korumak için çeşitli güvenlik mekanizmaları uygular.



Java'nın Özellikleri

- ❑ Java basit bir dildir
- ❑ Java nesne yönelimlidir
- ❑ Java dağıtıktır
- ❑ Java yorumlanabilirdir
- ❑ Java güçlü bir programlama dilidir
- ❑ Java güvenli bir programlama dilidir
- ❑ Java platformdan bağımsızdır.
- ❑ Java taşınabilirdir.
- ❑ Java işlevseldir.
- ❑ Java çoklu kullanım özelliği
- ❑ Java dinamiktir

Bir kere yaz her yerde çalıştır
(Write once, run anywhere)

Bir program yazdığınızda, (JVM) ile
her platformda çalıştırabilirsiniz.



Java'nın Özellikleri

- ❑ Java basit bir dildir
- ❑ Java nesne yönelimlidir
- ❑ Java dağıtıktır
- ❑ Java yorumlanabilirdir
- ❑ Java güçlü bir programlama dilidir
- ❑ Java güvenli bir programlama dilidir
- ❑ Java platformdan bağımsızdır.
- ❑ **Java taşınabilirdir.**
- ❑ Java işlevseldir.
- ❑ Java çoklu kullanım özelliği
- ❑ Java dinamiktir

Java platformdan bağımsız olduğu için her yere taşınabilir.



Java'nın Özellikleri

- ❑ Java basit bir dildir
- ❑ Java nesne yönelimlidir
- ❑ Java dağıtıktır
- ❑ Java yorumlanabilirdir
- ❑ Java güçlü bir programlama dilidir
- ❑ Java güvenli bir programlama dilidir
- ❑ Java platformdan bağımsızdır.
- ❑ Java taşınabilirdir
- ❑ **Java işlevseldir**
- ❑ Java çoklu kullanım özelliği
- ❑ Java dinamiktir

Taşınabilirlik ve platformdan bağımsız olması nedeniyle her platformda yeniden derlenebilir. Bu nedenle işlevseldir.



Java'nın Özellikleri

- ❑ Java basit bir dildir
- ❑ Java nesne yönelimlidir
- ❑ Java dağıtıktır
- ❑ Java yorumlanabilirdir
- ❑ Java güçlü bir programlama dilidir
- ❑ Java güvenli bir programlama dilidir
- ❑ Java platformdan bağımsızdır
- ❑ Java taşınabilirdir.
- ❑ Java işlevseldir.
- ❑ Java çok iş parçacıklı kullanım özelliği
- ❑ Java dinamiktir

Bazı dillerde bu özelliği kullanabilmek için bazı spesifik altyordamlar çağırmak gerekmektedir. Fakat multithread programming olarak bilinen bu özellik Java'ya entegre edilmiştir.

Java'nın Özellikleri

- ❑ Java basit bir dildir
- ❑ Java nesne yönelimlidir
- ❑ Java dağıtıktır
- ❑ Java yorumlanabilirdir
- ❑ Java güçlü bir programlama dilidir
- ❑ Java güvenli bir programlama dilidir
- ❑ Java platformdan bağımsızdır
- ❑ Java taşınabilirdir.
- ❑ Java işlevseldir.
- ❑ Java çoklu kullanım özelliği
- ❑ **Java dinamiktir**

Java gelişen çevreye uyum sağlamak için tasarlanmıştır. Yeni kod derleme yapmadan anında yüklenebilir. Geliştiricilerin çok büyük yeni yazılım sürümleri oluşturmalarına ve yüklemesine gerek yoktur. Yeni özellikler gerektiğinde dahil edilebilir.

Popüler Java Entegre Geliştirme Ortamları (JAVA IDEs)

- ❑ NetBeans (Açık kaynak kodlu, Sun)
- ❑ Eclipse (Açık kaynak kodlu ,IBM)
- ❑ IntelliJ IDEA (Açık kaynak kodlu, JetBrains)





Program Tasarımı & Problem Çözme Teknikleri

Program nasıl yazılır?

- Bilgisayar zeki değildir.
 - Bir problemi analiz edip çözüm üretemez.
 - İnsan (Programcı) problemi analiz eder, problem çözümü için komutlar geliştirir ve bilgisayarın bu komutları çalıştırmasını sağlar.
- Program yazabilmek için iki aşama gereklidir: **Problem çözme** aşaması ve **Uygulama** aşaması

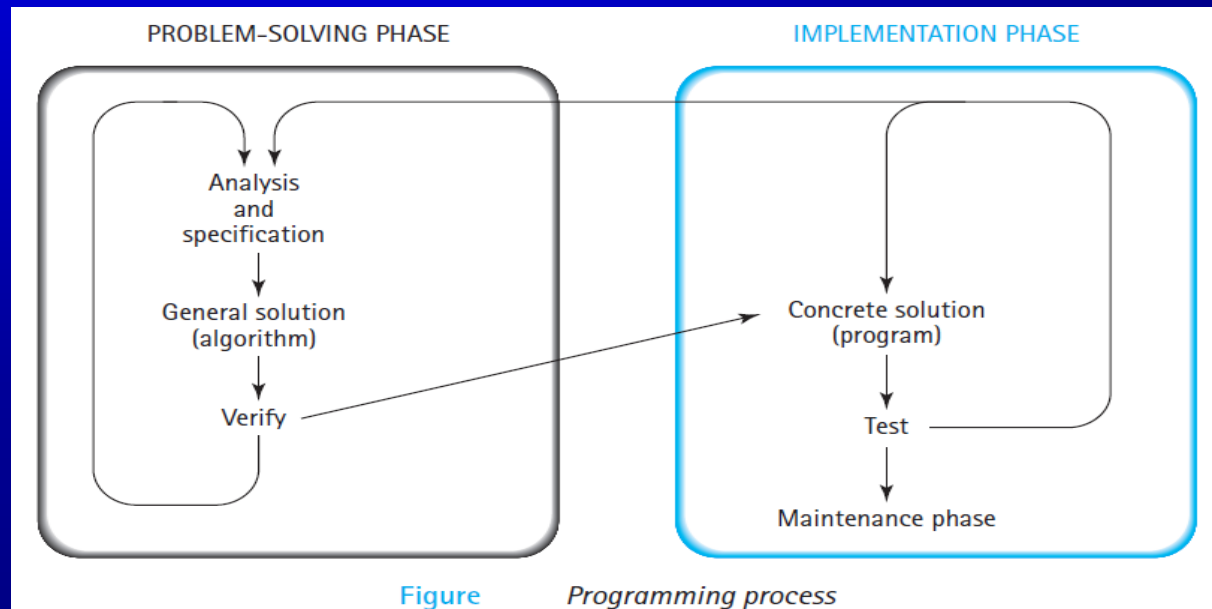


Figure Programming process

Problem-Çözme Aşaması

- **Analiz ve Özellikler (Analysis and Specification)** – Problemi anlama ve tanımlama, çözümünün ne olduğunu belirleme
- **Genel Çözüm (General Solution (Algorithm))**- Gerekli data tiplerini belirleme, problemi çözen adımların mantıksal olarak sıralanması
- **Doğrulama (Verify)**- Gerçekten algoritmanın problemi çözdüğünü görebilmek adına adımları tam olarak takip edip doğrulama



Uygulama Aşaması

- **Belirli Çözüm (Concrete Solution (Program))-**
Algoritmayı bir programlama diline dönüştürme
- **Test-**
 - Sonuçlar manuel olarak kontrol edilir.
 - Hata bulunursa, program ve algoritma analiz edilir hata kaynağı tespit edilir ve düzeltilir.
- Program test edildikten sonra bir sonraki aşamaya geçilir (maintenance).
- **Bakım (Maintenance)** değişen gereksinimleri karşılamak veya kullanım sırasında ortaya çıkan hataları düzeltmek için gereklidir





Program Geliştirme Adımları

Program Geliştirme Adımları

1. Problemi 3 bileşene ayırarak tanımlayınız
 - Girdiler
 - Çıktılar
 - Çıktıları üretebilmek için gerekli işlem adımları.



Program Geliştirme Adımları

2. Çözüm çerçevesini oluşturma
 - Problemi küçük adımlara ayrıştırma
 - Çözüm için bir anahat belirleme
3. Anahatı bir algoritmaya dönüştürme
 - Algoritma oluşturulur



Program Geliştirme Adımları

4. Algoritma doğruluk testi.
 - Program geliştirmede en önemli ve en çok unutulan adım
 - Mantık hataları erken aşamada tespit edilebilir ve düzeltilebilir.
5. Algoritmayı belirli bir programlama diline kodlama



Program Geliştirme Adımları

6. Programı bilgisayarda çalıştırma.
 - Bu adımda program derleyicisi kullanılır ve programcı kodu test eder
 - Syntax hataları (Syntax Errors)
 - Mantık hataları (Logic Errors)
7. Program belgeleme ve bakım





Algoritmalar & Akış Diyagramları

■ Algoritma nedir?

- Bir görevin yerine getirilmesine ilişkin adımları listeler (bir tarif gibi)
- Bir algoritma :
 - Açık, kesin ve net olmalı
 - Her durum için doğru sonuç vermeli
 - Sonlanmalı

■ Pseudocode nedir?

- Yapılandırılmış İngilizce (üst düzey bilgisayar dili gibi görünmek için resmileştirilmiş ve kısaltılmıştır)



Pseudocode & Algoritma

- **Örnek 1:** Bir öğrencinin geçme notunu ve geçip geçmediğini belirleyen bir algoritma yazınız. Geçme notu öğrencinin almış olduğu 4 notun ortalaması alınarak hesaplanmaktadır.



Pseudocode & Algoritma

Pseudocode:

- *Input: 4 not*
- *Ortalamayı hesaplama: 4 notu topla ve 4'e böl*
- *Eğer ortalama 50'nin altındaysa*
 - “KALDI” yaz*
 - değilse*
 - “GEÇTİ” yaz*



Pseudocode & Algoritma

□ Detaylı Algoritma

- Adım 1: Input N1,N2,N3,N4
- Adım 2: $NOT \leftarrow (N1+N2+N3+N4)/4$
- Adım 3: if($NOT < 50$)
 - Print “Kaldı”
 - else
 - Print “Geçti”
 - endif



Akış Diyagramı

- Bir bilgi sistemi veya programındaki işlem dizisinin grafiksel gösterimi
- Program akış çizelgeleri, tek bir programdaki veya alt programdaki komutların sırasını gösterir.
 - Algoritmanın mantığını gösterir
 - Her bir adımı ve bağlantılarını vurgular

Not: Her çeşit akış diyagramı için farklı semboller kullanılır.



Akış Diyagramı Sembolleri

Ad	Sembol	Akış diyagramındaki anlamı
Oval		Başlangıç ve bitiş
Paralelkenar		Girdi işlemi
Dikdörtgen		İşlem aşaması Örn: toplama ,bölme vb.
Eşkenar dörtgen		Karar/Karşılaştırma aşaması
Hybrid		Çıktı işlemi
Akış çizgisi		Akış yönünü gösterir



Örnek 1

- Metreyi santimetreye çeviren algoritmayı yazın ve akış diyagramını çiziniz Write an algorithm and draw a flowchart to convert the length in feet to centimeter.

Pseudocode:

- *Input : Uzunluk(m)*
- *İşlem:cm, m birimindeki uzunluğun 100 ile çarpılması ile elde edilir.*
- *Çıktı: cm türündeki uzunluk*

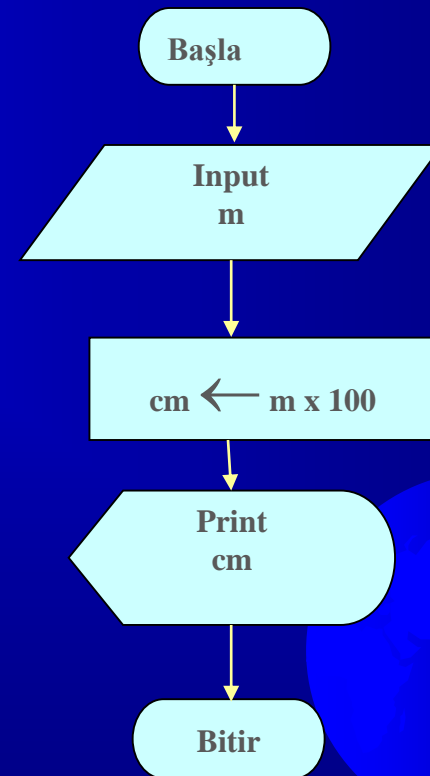


Örnek 1

Algoritma

- Adım 1: Input m
- Adım 2: $cm \leftarrow m \times 100$
- Adım 3: Print cm

Flowchart



Örnek 2

Kısa ve uzun kenar uzunlukları verilen bir dikdörtgenin alanını hesaplayan algoritmayı yazın ve akış diyagramını oluşturun

Pseudocode

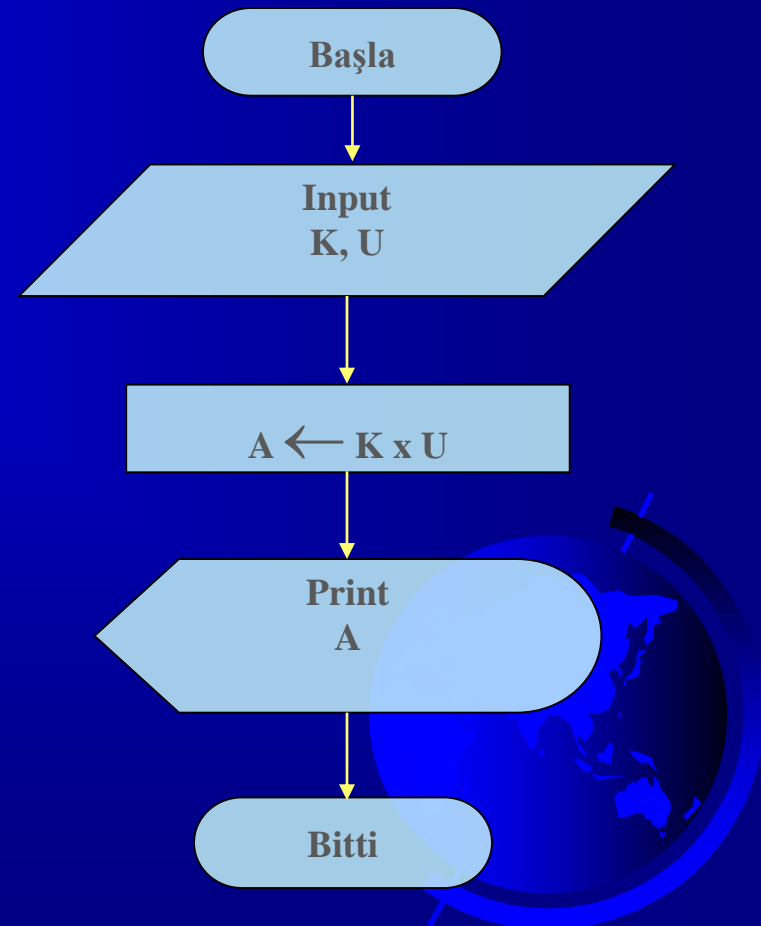
- *Input: Uzunluklar K ve U*
- *İşlem: Alan(A), K ve U 'nun çarpımı ile hesaplanır*
- *Çıktı: A*



Örnek 2

Algoritma

- Adım 1: Input W,L
- Adım 2: $A \leftarrow L \times W$
- Adım 3: Print A



Örnek 3

- Aşağıdaki denklemin köklerini bulan algoritmayı yazınız ve akış diyagramını çizersiniz

$$ax^2 + bx + c = 0$$

- İpucu: $d = \text{sqrt}(b^2 - 4ac)$, ve kökler: $x1 = (-b + d)/2a$ ve $x2 = (-b - d)/2a$



Örnek 3

Pseudocode:

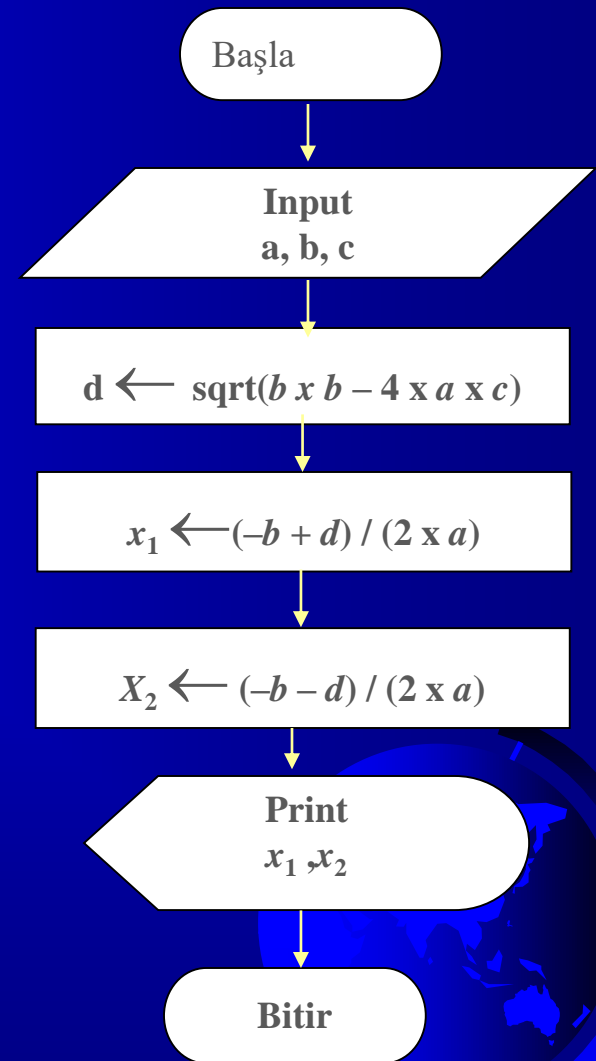
- *Input the coefficients (a , b , c) of the quadratic equation*
- *Calculate d*
- *Calculate x_1*
- *Calculate x_2*
- *Print x_1 and x_2*



Örnek 3

□ Algoritma:

- Adım 1: Input a, b, c
- Adım 2: $d \leftarrow \text{sqrt}(b \times b - 4 \times a \times c)$
- Adım 3: $x_1 \leftarrow (-b + d) / (2 \times a)$
- Adım 4: $x_2 \leftarrow (-b - d) / (2 \times a)$
- Adım 5: Print x1, x2



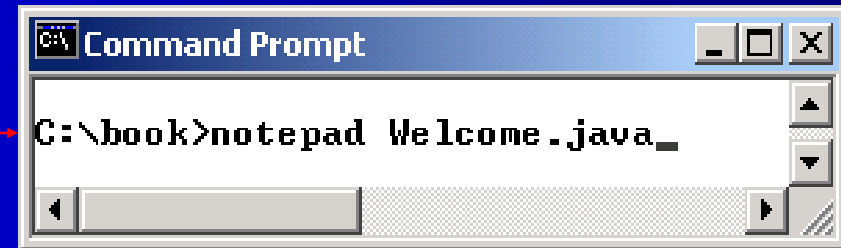
Basit Java Program1

```
//This program prints Welcome to Java!  
public class Welcome {  
    public static void main(String[] args) {  
        System.out.println("Welcome to Java!");  
    }  
}
```

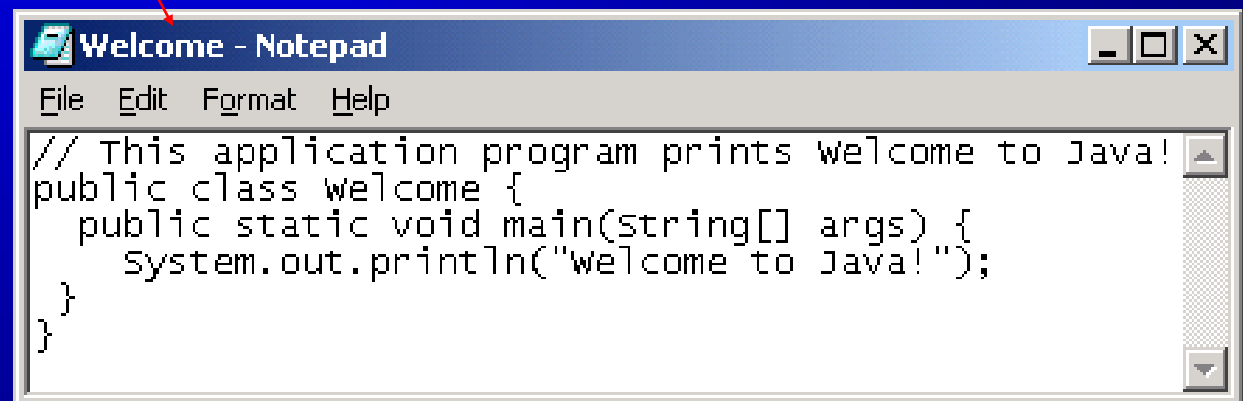


NotePad kullanarak kod yazma ve düzenleme

NotePad kullanırken,
komut satırına
notepad Welcome.java
yazınız



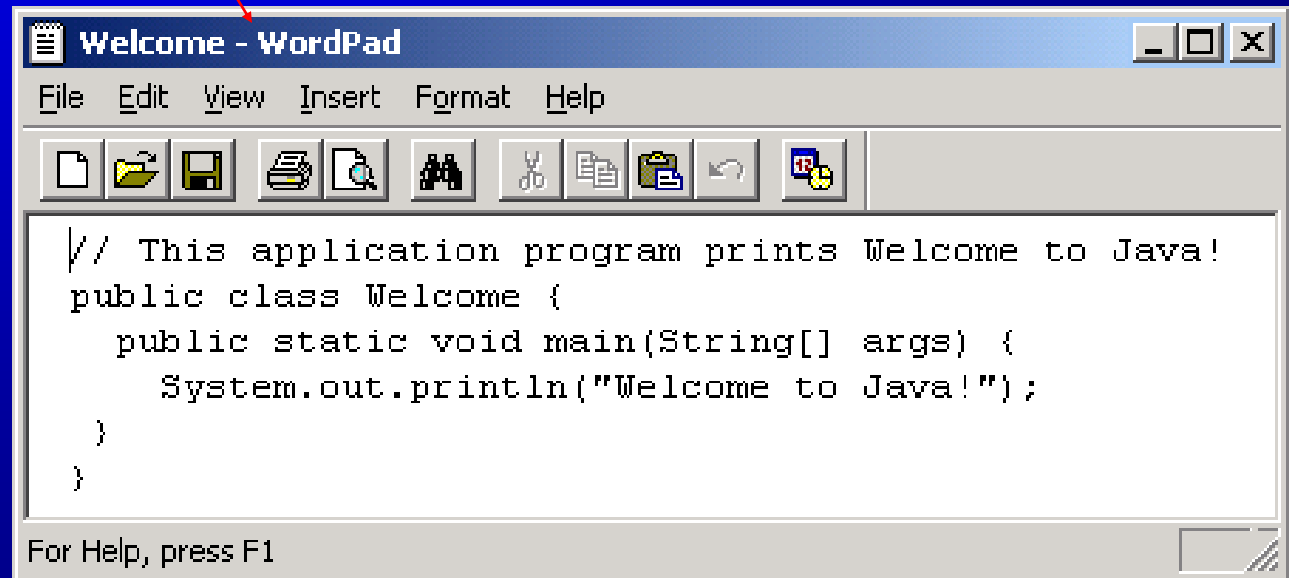
```
Command Prompt
C:\book>notepad Welcome.java_
```



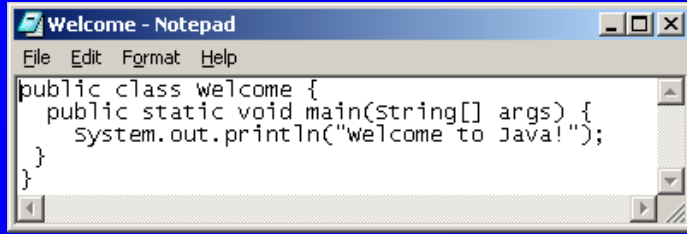
```
Welcome - Notepad
File Edit Format Help
// This application program prints welcome to Java!
public class Welcome {
    public static void main(string[] args) {
        system.out.println("welcome to Java!");
    }
}
```

WordPad kullanarak kod yazma ve düzenleme

WordPad veya notebook
kullanmak için,
write Welcome.java



Program Oluşturma, Derleme ve Çalıştırma



```
File Edit Format Help
public class welcome {
    public static void main(String[] args) {
        System.out.println("welcome to Java!");
    }
}
```

Kaynak kodu (Programcı tarafından
oluşturulur)

```
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```

Diske kaydedilir

Kaynak Kod

Byte code (JVM'nin okuyup anlayabilmesi için
derleyici tarafından üretilir)

```
...
Method Welcome()
  0 aload_0
  ...
Method void main(java.lang.String[])
  0 getstatic #2 ...
  3 ldc #3 <String "Welcome to
Java!">
  5 invokevirtual #4 ...
  8 return
```

Diskte depolanır

Bytecode

Çalıştır Bytecode
i.e., java Welcome

Sonuç

Kaynak kodunu
oluşturma/modifiye etme

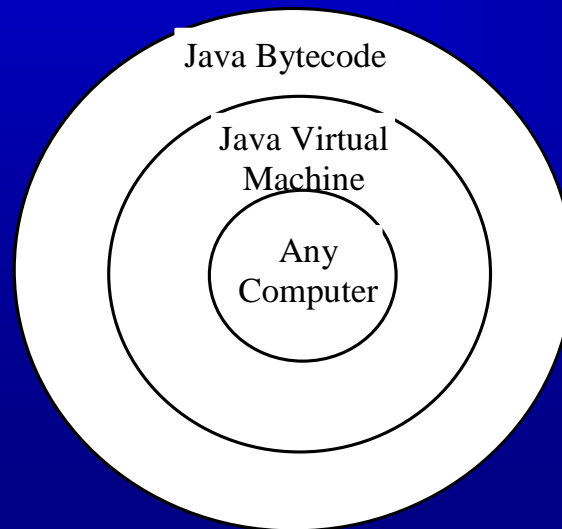
Kaynak kod derleme
i.e., javac Welcome.java

Eğer derleme hatası varsa

Eğer runtime hatası oluşursa veya
hatalı sonuç elde edilirse

Java kaynak kodunun derlenmesi

Bir kaynak programı uygun derleyicilere sahip herhangi bir makineye taşıyabilirsiniz. Ancak kaynak program yeniden derlenmelidir, çünkü nesne programı sadece belirli bir makinede çalışabilir. Günümüzde bilgisayarlar birlikte çalışmak için ağa bağlanmıştır. Java, herhangi bir platformda nesne programlarını çalıştırmak için tasarlanmıştır. Java ile programı bir kere yazıp, kaynak programı bytecode olarak bilinen özel bir nesne kodu türünde derlersiniz. Bytecode daha sonra aşağıda gösterildiği gibi Java Sanal Makinesi olan herhangi bir bilgisayarda çalışabilir. Java Sanal Makinesi, Java bytecode'unu yorumlayan bir yazılımdır.



Program Çalıştırılması

Enter main method

```
//This program prints Welcome to Java!  
public class Welcome {  
    public static void main(String[] args) {  
        System.out.println("Welcome to Java!");  
    }  
}
```

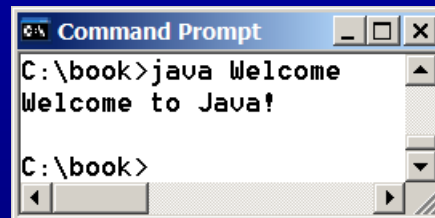

Program Çalıştırılması

Execute statement

```
//This program prints Welcome to Java!  
public class Welcome {  
    public static void main(String[] args) {  
        System.out.println("Welcome to Java!");  
    }  
}
```

Program Çalıştırılması

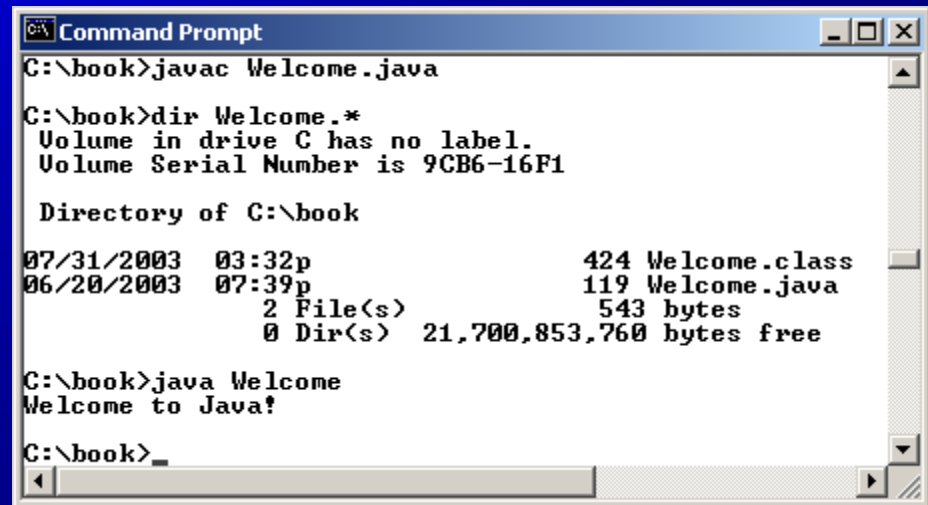
```
//This program prints Welcome to Java!  
public class Welcome {  
    public static void main(String[] args) {  
        System.out.println("Welcome to Java!");  
    }  
}
```



print a message to the console

Komut satırında Java kodunun derlenmesi ve çalıştırılması

- JDK bin dosyasının yolunu ayarlama (Set path to JDK bin directory)
 - set path=c:\Program Files\java\jdk1.6.0\bin
- Bulunduğun dizine sınıf yolunu getirme
 - set classpath=.
- Derleme
 - javac Welcome.java
- Çalıştırma
 - java Welcome



```
Command Prompt
C:\book>javac Welcome.java

C:\book>dir Welcome.*
Volume in drive C has no label.
Volume Serial Number is 9CB6-16F1

Directory of C:\book

07/31/2003  03:32p                424 Welcome.class
06/20/2003  07:39p                119 Welcome.java
                2 File(s)            543 bytes
                0 Dir(s)  21,700,853,760 bytes free

C:\book>java Welcome
Welcome to Java!

C:\book>
```

Java Programının Anatomisi

- Açıklama Satırları (Comments)
- Ayrılmış Kelimeler (Reserved words)
- Değiştiriciler (Modifiers)
- İfadeler (Statements)
- Bloklar (Blocks)
- Sınıflar (Classes)
- Metodlar (Methods)
- main method



Açıklama Satırları

Java'da 3 farklı şekilde açıklama yapılabilir:

Satır Açıklaması: (//)

Paragraf Açıklaması: / ... */*

*javadoc açıklaması: /** ... */ Belgeleme için kullanılır*



Ayrılmış Kelimeler

Ayrılmış kelimeler veya anahtar kelimeler, derleyici için belirli bir anlamı olan ve programdaki diğer amaçlar için kullanılamayan kelimelerdir.

Örn: public, static, void.



Değiştiriciler

Java, verilerin özelliklerini, yöntemlerini ve sınıflarını ve bunların nasıl kullanılabileceğini belirten değiştiriciler adı verilen belirli sözcükleri kullanır.

❖ public, static, ...



İfadeler

Bir ifade, bir eylemi veya bir dizi eylemi temsil eder. Her ifade Java'da (;) ile biter.



Bloklar

Bir programdaki {...} aralığı, bir programın bileşenlerini gruplayan bir blok oluşturur.

```
public class Test {  
    public static void main(String[] args) {  
        System.out.println("Welcome to Java!");  
    }  
}
```

Class block

Method block



Sınıf

- ❑ Sınıf, temel Java yapısıdır.
- ❑ Sınıf, nesneler için bir şablon veya plandır.
- ❑ Java'da programlamak için sınıfları anlamamız ve yazıp kullanabilmeniz gerekir.
- ❑ İlk aşamada kodlar main metodu içerisine yazılacak nesneye yönelik programlama ile ayrıntılara daha sonra girilecektir.



Metod

- ❑ Metodlar, karmaşık programları küçük yapılara bölerek daha anlaşılır ve yönetilebilir seviyeye getirir.
- ❑ Metod bir programın ayrılmış küçük parçalarıdır.
- ❑ Metod, konsolda bir mesajı görüntülemek için bir dizi işlem gerçekleştiren ifadeler topluluğudur. Nasıl çalıştığının ayrıntılarını tam olarak anlamadan da kullanılabilir.



main Method

main metodu, program akışının kontrolünü sağlar. Java yorumlayıcısı, uygulamayı main metodunu çağırarak yürütür.

```
public static void main(String[] args) {  
    // Statements;  
}
```



Basit Program Yapısı

