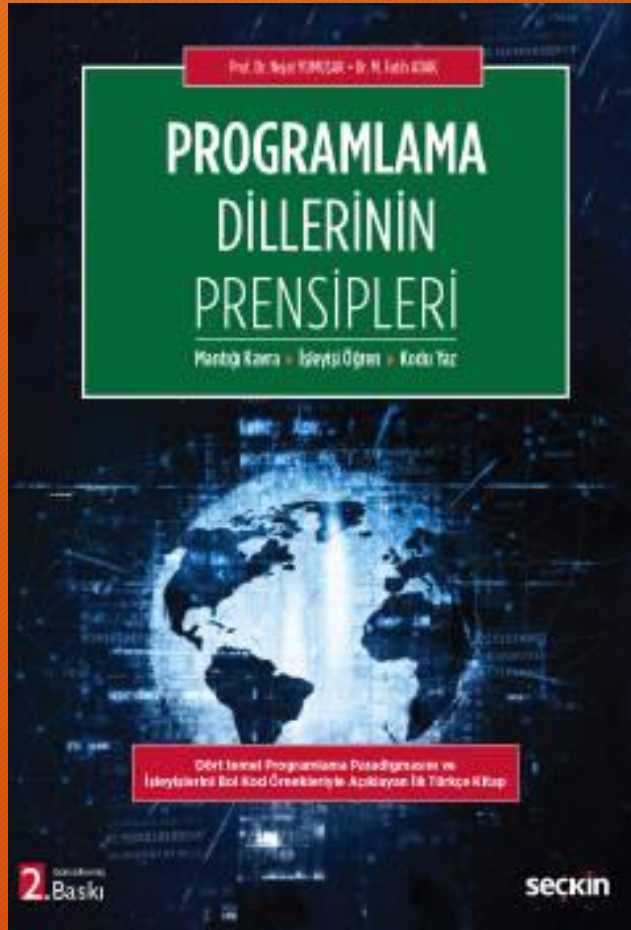


# Programlama Dillerinin Prensipleri

## Hafta 1 - Giriş

Dr. Öğr. Üyesi M. Fatih ADAK

# Ders Kitabı



<https://www.seckin.com.tr/kitap/989255263>

# Ders Akışı

Hafta	Konular
1	Giriş ve Programlama Kavramlarının Tanıtılması
2	Dillerin tarihçesi ve evrimi
3	Dil tanımlanması ve Dil çevrimi
4	Temel programlama kavramları, Veri tipleri ve yapıları
5	Bağlama Kavramları ve İsim Kapsamları
6	Yapısal programlama
7	Altprogramlar ve Modülasyon
8	Parametre aktarım yöntemleri
9	Nesne yönelimli programlama kavramları
10	Programlama dillerinde hata yakalama
11	Programlama dillerinde eşzamanlılık
12	Yorumlamalı Diller ve Python
13	Fonksiyonel programlama kavramları
14	Mantıksal programlama kavramları



# Değerlendirme Sistemi

- 1. Ödev (Java) : % 19.8
- 2. Ödev (C) : % 19.8
- Kısa Sınav (Test) : % 5.4
- Ara Sınav (Klasik) : % 15
- Final (Test-Klasik) : % 40
  
- Ödevler bireyseldir.

# Ödev ve Kod Derslerinde Araçlar

- Java için Eclipse 2021-12
  - <https://www.eclipse.org/downloads/>
- C Dili için
  - MinGW
  - <http://www.mingw.org/>
- Lisp Dili için
  - GNU Common Lisp 2.6.12
  - <https://www.gnu.org/software/gcl/>

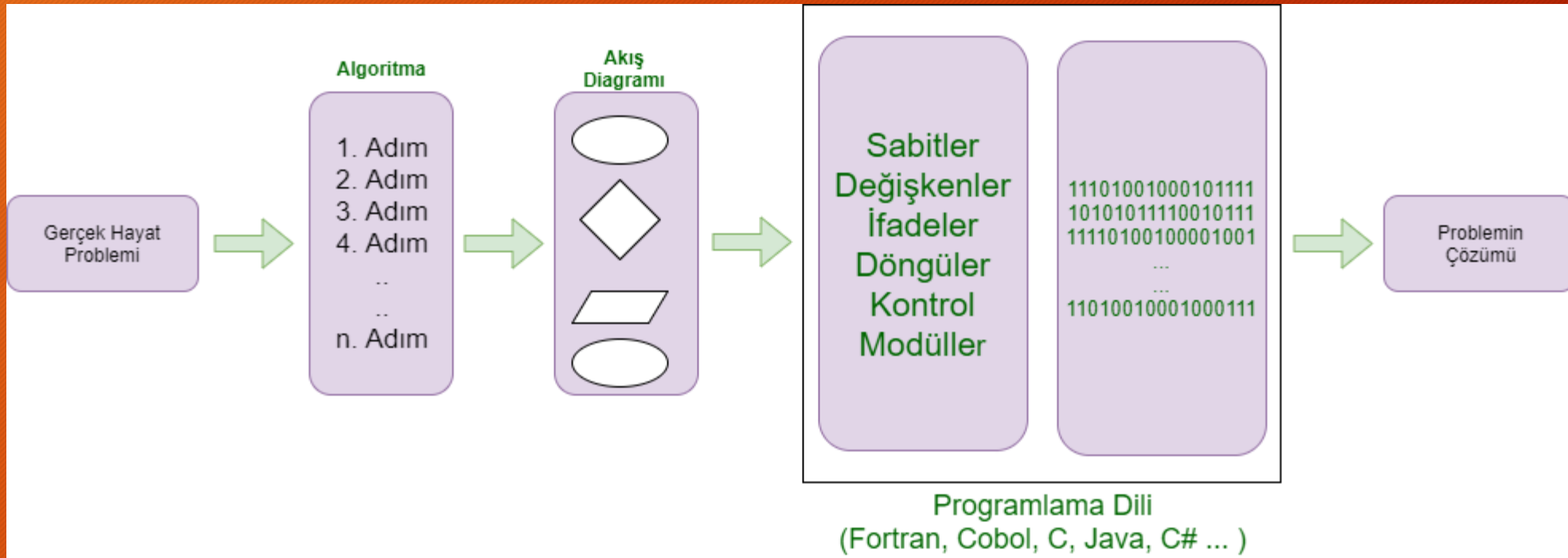
# Bu Haftaki İçerik

- Programlama Dillerinin Sınıflandırılması
- Programlama Dili Paradigmaları
- Programlama Dillerinin Değerlendirme Ölçütleri
- C Dili
- Java Dili
- Lisp Dili
- Prolog



# Programlama Dili Nedir?

- Bir problemin çözümünün bilgisayardaki gerçekleştirimini ifade etmek üzere program oluşturmak için kullanılan araca denir.



# Yazılımın Uygulama Alanlarına Göre Gruplandırılması

- Bilimsel ve mühendislik yazılımları
- Mesleki yazılımlar
- Yapay zeka yazılımları
- Görüntüsel yazılımlar
- Sistem yazılımları



# Programlama Dilinin Amaçları

- Bir programlama dili makinalara talimat vermek için gerekli bir araçtır.
- Programcılar arasında iletişim için gerekli bir vasıtaadır.
- Yüksek seviyeli tasarımları ifade etmek için gerekli bir araçtır.
- Algoritmaları göstermeye yarayan bir notasyondur.
- Genel kavramlar arasındaki yakınlıkları ifade etmeye yarayan bir yoldur.
- Çözümlerin ve çözüm yollarının test edilmesi için gerekli bir araçtır.
- Bilgisayarlı cihazları kontrol etmek için gerekli bir vasıtaadır.

# Programlama Dillerinin Sınıflandırılması

- Seviyelerine Göre Sınıflandırma
- Uygulama Alanlarına Göre Sınıflandırma
- Tasarım Paradigmalarına Göre Sınıflandırma

# Programlama Dillerinin Seviyelerine Göre Sınıflandırma

- Makine Dili
- Alçak Seviyeli Diller
- Orta Seviyeli Diller
- Yüksek Seviyeli Diller
- Çok Yüksek Seviyeli Diller



# Makine Dili

- Bir bilgisayarın doğrudan anlayabildiği bir dildir.
- Bilgisayarın ana dili olarak kabul edilir.
- Makine dili taşınabilir değildir ve makineye özgü yazılması gerekir.
- Makine dilinde kod yazmak çok zahmetli, çok zaman alıcı ve uğraştırıcıdır.

# Alçak Seviyeli Programlama Dili

- Sembolik Dil (Assembly)

- Makine dili kullanımının getirdiği problemleri ortadan kaldırmak üzere yapılan çalışmalarda
- Önce makine dilinin anlaşılma zorluğunu kısmen de olsa ortadan kaldırmak üzere sembolik dil geliştirilmiştir.
- Sembolik dilde 0 ve 1'ler yerine İngilizce ifadeler yer almaktaydı.
- Burada bellekten okuma yazma yerine çok daha hızlı olması açısından register'lar kullanılır.

Sembolik dilde ekrana Merhaba yazdırılması
<pre>mesaj db 'Merhaba', 0x0d, 0x0a, '\$' mov dx, mesaj mov ah, 9 int 0x21</pre>

# Orta Seviyeli Diller

- Sembolik diller bilgisayar kullanımını hızla arttırmıştır.
- Ancak çok basit işlemler için bile birçok komut gerekmektedir.
- Ayrıca sembolik diller her seferinde makine diline çevrilip öyle çalıştırılıyordu. Bu işlem program hızını 30 kat yavaşlatıyordu.
- Grace Hopper, bu problemin çözümü için derleyici fikrini ortaya attı.
- Program kodu bir kez derlenip makine diline çevrilecek ve bir daha bu işleme gerek kalmayacaktı.



# Orta Seviyeli Diller

- Ada, C gibi diller örnek verilebilir.
- Daha az kayıpla makine diline çevrilebildiğinden daha hızlı çalışır.
- Program yazmak yine zordur fakat sembolik dile göre oldukça kolaydır.

# Yüksek Seviyeli Diller

- Fortran ilk yüksek seviyeli dildir.
- Program yazmak daha kolay fakat orta seviyeli dillere göre program hızı daha yavaştır.
- Bu seviyedekiler 3. kuşak diller olarak kabul edilir.

# Çok Yüksek Seviyeli Diller

- Genellikle algoritmik yapı içermeyen görsel bir ortamda yazılan dillerdir.
- 4. Kuşak olarak isimlendirilirler.
- Java, C#, Visual Basic, Access, Oracle Forms bu seviyeye örnek verilebilir.
- Program hızları makine dillerine göre oldukça yavaştır.



# Programlama Dillerini Uygulama Alanlarına Göre Sınıflandırma

- Bilimsel ve Mühendislik Uygulama Dilleri
  - Pascal, C, Fortran
- Veritabanı Dilleri
  - MSSQL, Oracle Forms, XBASE
- Genel Amaçlı Programlama Dilleri
  - Pascal, C, Basic, Java
- Yapay Zeka Dilleri
  - Prolog, Lisp
- Modelleme Yapmak Üzere Geliştirilen Simülasyon Dilleri
  - GPSS, Simula67

# Programlama Dillerini Uygulama Alanlarına Göre Sınıflandırma

- Makro Diller (Script Diller)
  - awk, Perl, Python, Tcl, Javascript
- Sistem Programlama Dilleri
  - C (UNIX işletim sisteminin %80'i C dili ile geri kalanı sembolik dil ile yazılmıştır.)
- Ticari Uygulamalara Yönelik Programlama Dilleri
  - Cobol

# Programlama Dillerini Tasarım Paradigmalarına Göre Sınıflandırma

- Emir Esaslı Programlama
- Nesneye Yönelik Programlama
- Fonksiyonel Programlama
- Mantık Esaslı Programlama



# Programlama Dillerini Tasarım Paradigmalarına Göre Sınıflandırma

- Emir Esaslı (Impretive) Programlama

Deyim 1
Deyim 2
.
.
.
Deyim n

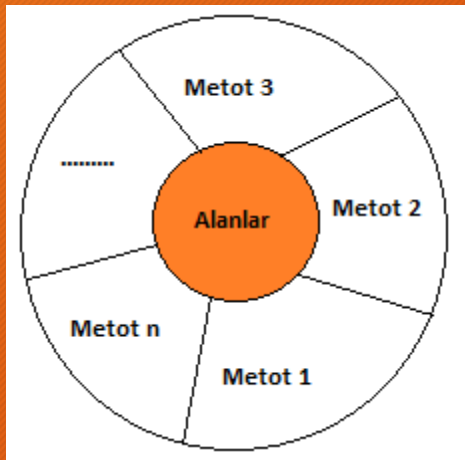
- Emir esaslı programlama dilleri işlem tabanlı olup, bir program bir dizi işlem olarak görülür.
- Bu diller yaygın olarak kullanılan ilk dil grubudur.
- C, Fortran, Pascal, Cobol örnek olarak verilebilir.

- Örneğin, atama işlemi bir deyimdir.

Farklı dillerde atama işlemleri
C dili x=13
Pascal dili x:=13
APL dili x<--13
Scheme veya Lisp dili (setq x 13)

# Programlama Dillerini Tasarım Paradigmalarına Göre Sınıflandırma

- Nesneye Yönelik Programlama



- Temeli simula67 programlama dilidir.
- Nesnelerin sınıfı ve alt sınıflara gruplanması, nesneye yönelik programlamanın temel noktasıdır.
- Karmaşık veri nesneleri ve bu veriler üzerinde çalışacak işlemler (metotlar) tasarlanır.



Nesneye yönelik programlamanın genel yapısı

# Programlama Dillerini Tasarım Paradigmalarına Göre Sınıflandırma

- Fonksiyonel Programlama

- Veriler ve sonucu elde etmek için veriye uygulanacak fonksiyonel dönüşümler bu paradigmanın temelini oluşturur.
- Lisp, Scheme ve ML dilleri bu paradigmaya örnektir.



- Fonksiyonel programlamanın temelini oluşturan parçalar





# Programlama Dillerini Tasarım Paradigmalarına Göre Sınıflandırma

- Mantık Esaslı Programlama
  - Bir işin nasıl yapılacağıının belirtilmesi yerine, ne yapılması istendiğinin belirtilmesi olarak görülür.
  - Belirli bir koşulun varlığını kontrol ederek ve koşul sağlanıyorsa, uygun bir işlem gerçekleştirerek çalışırlar.
  - Emir esaslı programlamaya benzer fakat deyimler sıralı olarak işlenmez.

Mantıksal paradigmayı destekleyen dillerin sözdizimi
Şart_1 → Hareket_1
Şart_2 → Hareket_2
Şart_3 → Hareket_3
...
...
...
Şart_n → Hareket_n

# Programlama Dillerinin Değerlendirme Ölçütleri

- Her programlama dili bir düşünce biçimi olduğundan binlerce programlama dili vardır denilebilir.
- Çözülecek problemin tipine ve uygulama alanına göre programlama dilleri arasında seçim yapmak için çeşitli değerlendirme ölçütlerine ihtiyaç duyulmaktadır.
- Her alan için en iyi olan bir programlama dili yoktur.

# Programlama Dillerinin Değerlendirme Ölçütleri

- İfade Gücü (Expression Power)
- Veri Türleri ve Yapıları (Data Types and Structures)
- Giriş/Çıkış Kolaylığı (Input/Output Facilities)
- Taşınabilirlik (Portability)
- Alt programlama Yeteneği (Modularity)
- Verimlilik (Efficiency)
- Okunabilirlik (Readability), Yazılabilirlik
- Esneklik (Flexibility)
- Öğrenme Kolaylığı (Pedagogy)
- Genel Amaçlılık (Generality)
- Yapısallık (Structurulness)
- Nesne yönelimlilik (Object Orientation)

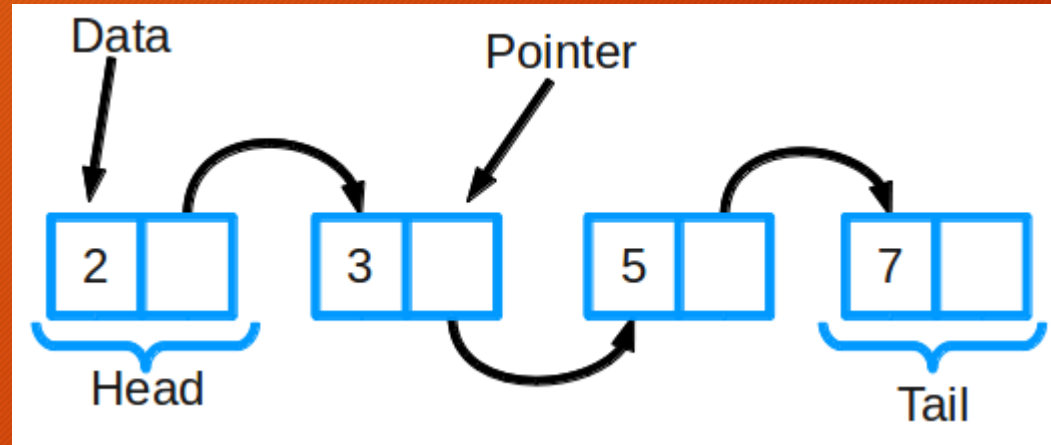


# Programlama Dillerinin Değerlendirme Ölçütleri

- İfade Gücü (Expression Power)
  - Algoritmayı tasarlayan kişinin niyetlerini açık bir biçimde yansıtabilmesine olanak tanınması
  - Günümüz popüler programlama dillerinin ifade gücü yüksektir.

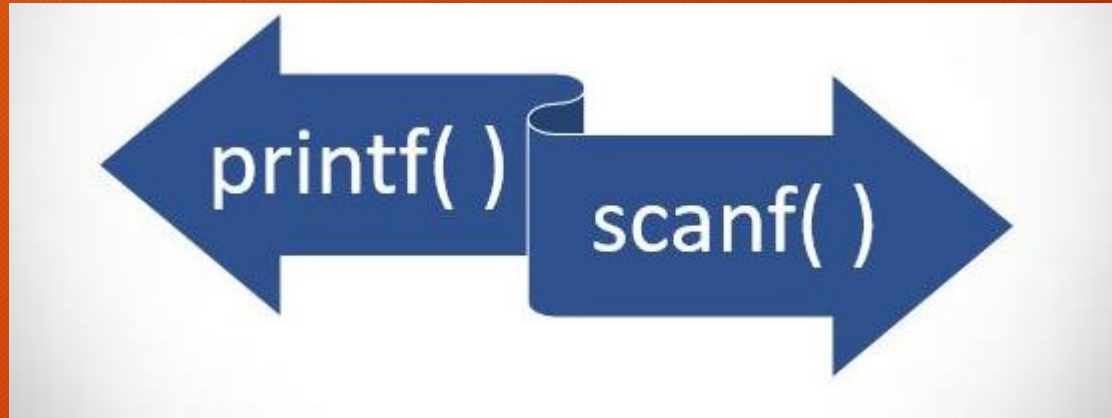
# Programlama Dillerinin Değerlendirme Ölçütleri

- Veri Türleri ve Yapıları (Data Types and Structures)
  - Çeşitli veri türlerini (tamsayı, gerçek sayı, karakter...) ve veri yapılarını (diziler, bağlı liste, kuyruk, yapılar vs.) destekleme yeteneğidir.



# Programlama Dillerinin Değerlendirme Ölçütleri

- Input Output Kolaylığı
  - Program yazmayı kolaylaştıran ve ifade gücünü arttıran bir özelliktir
  - Örneğin C dilinde bu ölçüt çok yüksek değildir.

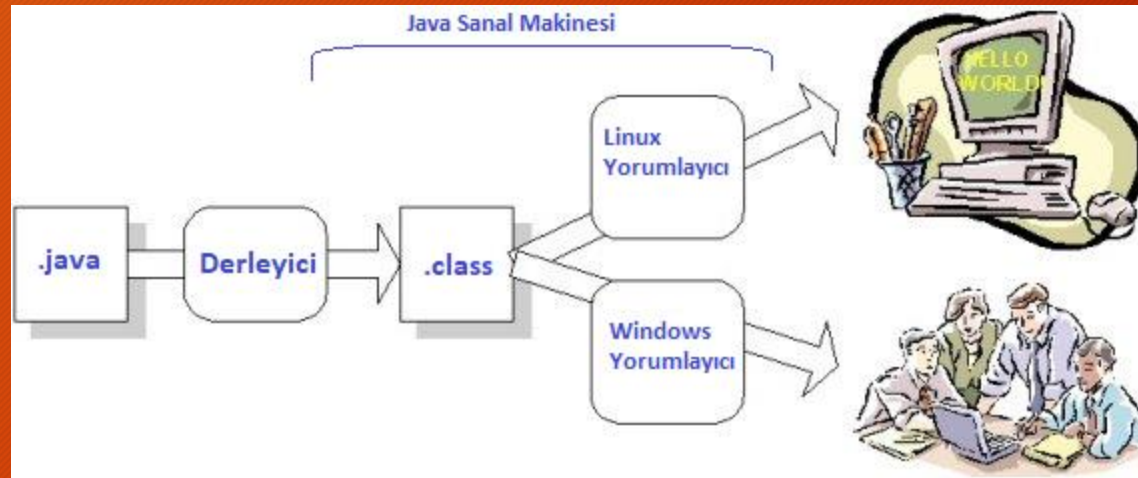




# Programlama Dillerinin Değerlendirme Ölçütleri

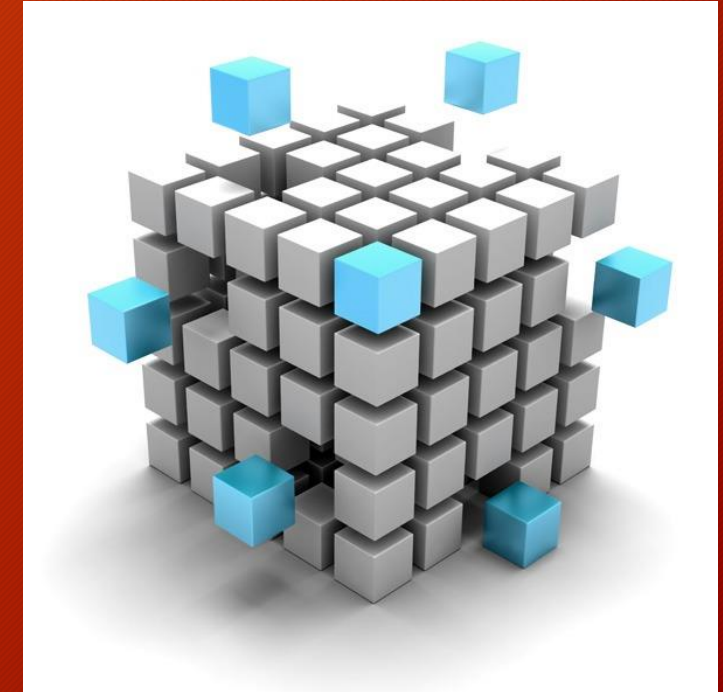
- Taşınabilirlik

- Bu terim kaynak kod için kullanılır.
- Bir sistemde yazılmış kaynak kodun bir başka sistemde de sorunsuz derlenip çalışmasıdır.
- Dillerin seviyesi düştükçe taşınabilirlik azalır.
- Hiçbir dil için mükemmel taşınabilirlik mümkün değildir.



# Programlama Dillerinin Değerlendirme Ölçütleri

- Alt Programlama Yeteneği
  - Bir programı parçalar halinde yazmayı desteklemesidir.
  - Yapısal Programlama tekniğinin vazgeçilmez bir parçasıdır.
  - Yazılacak kodu oldukça azaltır.
  - Program kodlarının anlaşılmasını kolaylaştırır.





# Programlama Dillerinin Değerlendirme Ölçütleri

- Verimlilik
  - Amaç koda dönüştürülmüş programların hızlı çalışabilmesine verimlilik denir.
  - Verimlilik derleyici, dil seviyesi ve dilin genel yapısına bağlıdır.
  - Çalışabilir kodun küçüklüğü ile çalışma hızı arasında doğrusal bir ilişki vardır.
  - C programları hızlı çalışır ve az yer kaplar.



# Programlama Dillerinin Değerlendirme Ölçütleri

- Okunabilirlik
  - Kaynak kodun çabuk ve kolay bir biçimde algılanabilmesi anlamına gelir.
  - Okunabilirlik güncelleştirmeyi kolay kılar ve proje grubu halinde kodun üzerinde çalışılabilmesine olanak sağlar.
  - En iyi program kodu **anlaşılamayan ama çok zekice yazılmış kod değildir**.
  - En kolay okunabilen ve anlaşılabilen kod en iyi koddur.

"Babaannene iki dakikada açıklayamayacağın tek bir satır kodu bile programına ekleme, hatta babaannen Ada Lovelace olsa bile."

*Anonim*



# Programlama Dillerinin Değerlendirme Ölçütleri

- Esneklik
  - Esneklik programlama dilinin programcıyı kısıtlamamasıdır.
  - Esnek dillerde birçok işlem, programcı için serbest bırakılmıştır.
  - Bu deneyimsiz programcılar için hata yapma riskini arttırır.
  - C esnek bir dil iken Java esnekliği çok kısıtlanmış bir dildir.

C dilinde karakter ve tamsayıların birbirine kolayca atanabilmesi

```
int main(){
    int x=97;
    char c=x;
    char b='b';
    int y=b;
    printf("%c\n",c);
    printf("%d\n",y);
    return 0;
}
```



# Programlama Dillerinin Değerlendirme Ölçütleri

- Öğrenme Kolaylığı
  - Programlama dillerinin seviyesi arttıkça öğrenme kolaylığı artar.
  - Yüksek seviyeli dillerin popüler olması öğrenme kolaylığına bağlıdır.
  - C dili öğrenmesi zor bir dil iken Java aksine basittir.



# Programlama Dillerinin Değerlendirme Ölçütleri

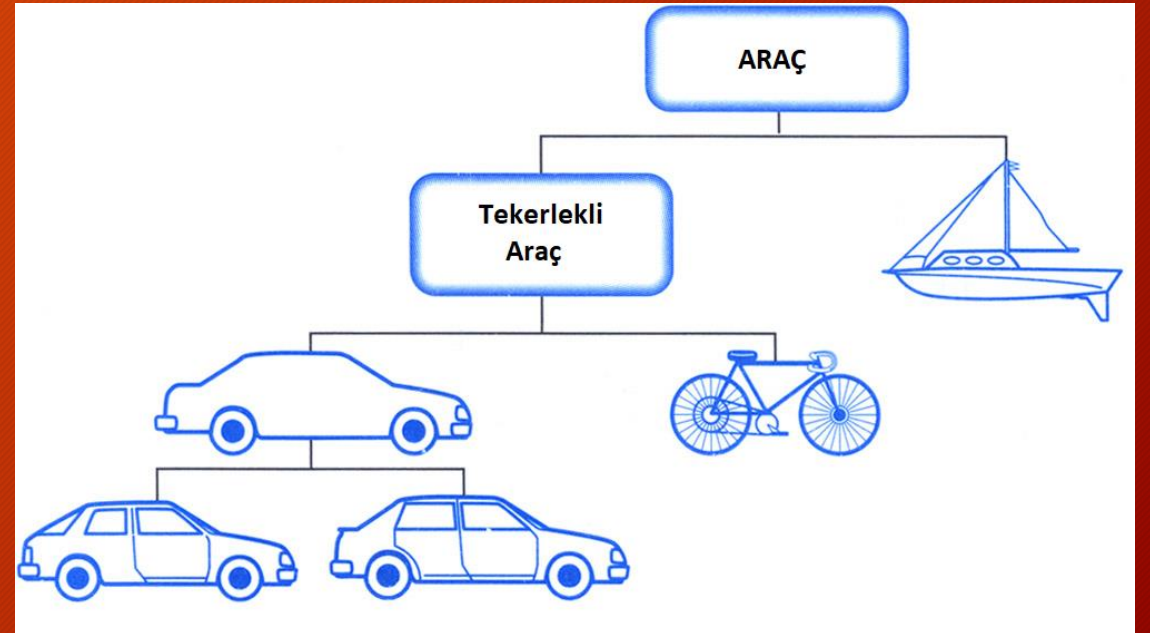
- Genel Amaçlılık
  - Programlama dillerinin çok çeşitli uygulamalarda etkin olarak kullanılabilmesidir.
  - Cobol dili ticari uygulamalarda etkin bir dil iken mühendislik uygulamalarında tercih edilmez.
  - Java dilinin genel amaçlılığı yüksektir.

# Programlama Dillerinin Değerlendirme Ölçütleri

- Yapısallık
  - Yapısallık bir programlama tekniğidir.
  - Bu tekniği kullanan dillerde bloklar halinde yazım ön plandadır.
  - Alt programlar yoğun olarak kullanılır.
  - C dili iyi bir örnektir.
  - Yapısal programlama 4 ana ilke üzerine kurulmuştur.
    - Böl ve Yönet
    - Veri Gizleme (lokal değişkenler)
    - Tek Giriş ve Çıkış
    - Döngüler ve Diğer Kontrol Yapıları

# Programlama Dillerinin Değerlendirme Ölçütleri

- Nesne Yönelimlilik
  - Veri + Program = Nesne
  - Büyük programların yazılması için tasarlanmış bir tekniktir.
  - Üç temel üzerine kurulmuştur.
    - Kapsülleme
    - Çok Biçimlilik
    - Kalıtım





# C Dili

- C dili ilk olarak Dennis Ritchie tarafından 1972 yılında Bell laboratuvarında geliştirilmiştir.
- C dili işletim sistemi dili olarak bilinir.
- C dili emir esaslı ve yapısal bir dildir.

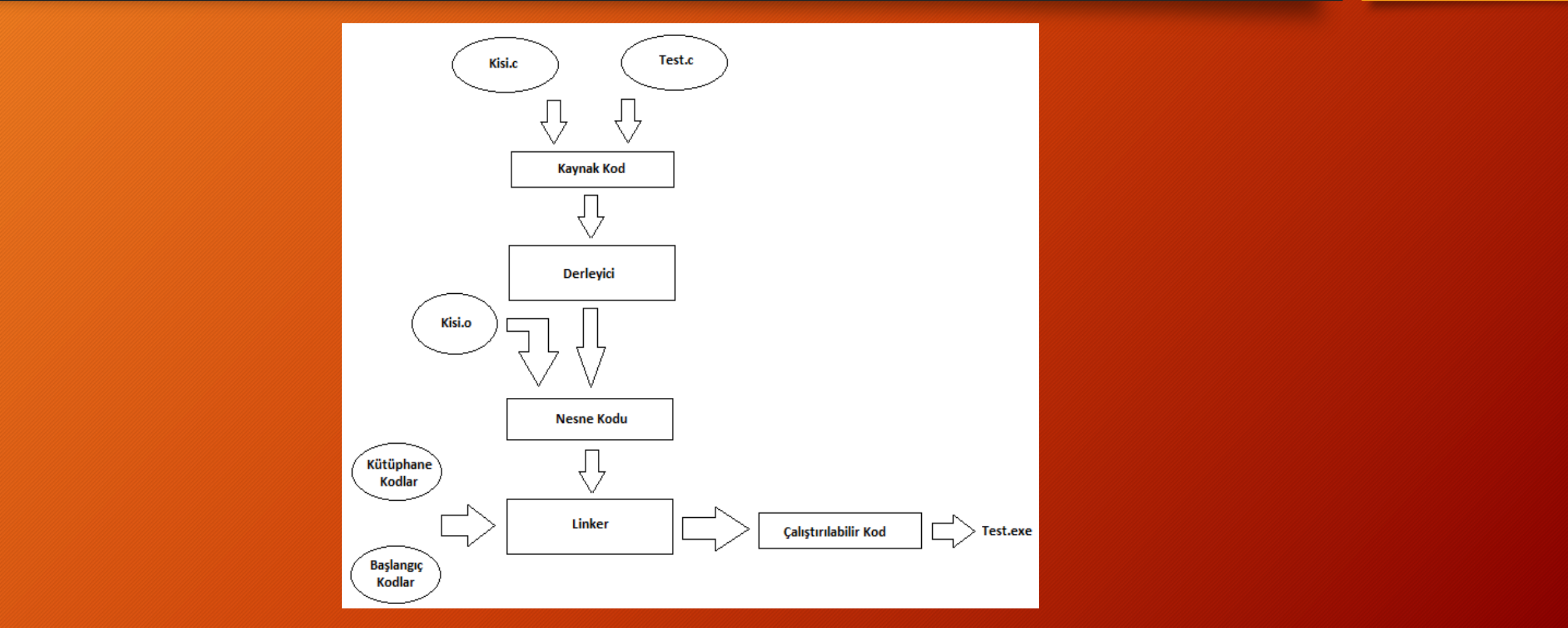
## Örnek bir C kodu

```
#include "stdio.h"
int main(){
    int yas;
    printf("Lutfen yasinizi girin:");
    scanf("%d",&yas);
    int dogumyili = 2017-yas;
    printf("Dogum Yiliniz:%d\n",dogumyili);
    return 0;
}
```

# make Dosyası

- C dilinde gelişmiş programların tasarlanmasında birden fazla kaynak kod ve başlık dosyası kullanılabilir. Bu durumda komutların tek tek el ile her seferinde komut satırına girilmesi zahmetli ve zaman alan bir iştir.
- Bu komutların bir dosyaya yazılıp dosyanın komut satırından çağırılması zamandan kazanç sağlayacaktır.
- Bu dosyanın ismine make dosyası (makefile) denilmektedir.

Örnek C kodunu derlemek için make dosyası
hepsi: derle calistir
 derle: gcc -I ./include/ -o ./lib/Kisi.o -c ./src/Kisi.c gcc -I ./include/ -o ./bin/Test ./lib/Kisi.o ./src/Test.c
 calistir: ./bin/Test





# Java Dili

- İlk olarak Oak ismiyle tasarlanmış ve gömülü sistemler için kullanılmıştır.
- Daha sonraları Java ismini alıp internet uygulama geliştirme için kullanılmıştır.
- Genel bir programlama dili olan Java her platformda kullanılabilir.
- Tamamen nesne yönelimlidir. En ufak program için yine Sınıf yazılmalıdır.

# Lisp Dili

- Sembolik veri işleme için tasarlanan Lisp dili fonksiyonel programlama dilidir.
- Daha çok yapay zeka çalışmalarında kullanılmıştır.
- İki temel veri yapısı içerir bunlar **Atom** ve **Liste** dir.

Ekrana merhaba yazan Lisp fonksiyonu
(Defun MerhabaYaz() "Merhaba" )

# Prolog Dili

- 1970 yılında İngiliz ve Fransız ortaklığında geliştirilmiştir.
- Mantıksal programlama dili olan Prolog'ta bildirme esaslı bir yapı kullanılır.
- Prolog'u iki temel konsept oluşturur.
  - Olaylar
    - Doğru olan durumlardan oluşur.
  - Kurallar
    - Kuralları ifade etmek için önermelerden faydalanılır.
      - Atomik önermeler
      - Bileşik terimler



# Kaynaklar

- Yumusak N., Adak M.F. *Programlama Dillerinin Prensipleri*. 2. Baskı, Seçkin Yayıncılık, 2021
- Sebesta, Robert W. *Concepts of programming languages*. 11 ed. Pearson Education Limited, 2016.
- Sethi, Ravi. *Programming languages: concepts and constructs*. Addison Wesley Longman Publishing Co., Inc., 1996.
- Watt, David A. *Programming language design concepts*. John Wiley & Sons, 2004.
- Malik, D. S., and Robert Burton. *Java programming: guided learning with early objects*. Course Technology Press, 2008.
- Waite, Mitchell, Stephen Prata, and Donald Martin. *C primer plus*. Sams, 1987.
- Hennessey, Wade L. *Common Lisp*. McGraw-Hill, Inc., 1989.
- Liang, Y. Daniel. *Introduction to Java programming: brief version*. pearson prentice hall, 2009.
- Yumusak N., Adak M.F. *C/C++ ile Veri Yapıları ve Çözümlü Uygulamalar*. 2. Baskı, Seçkin Yayıncılık, 2016