



YOZGAT BOZOK ÜNİVERSİTESİ

Programlama Dilleri

Prensipleri

Dr. Öğr. Üyesi R. Sinan ARSLAN



Programlama Dilleri Prensipleri

- Değerlendirme
 - Ara sınav: %40
 - Ödevler: %30
 - Final Sınavı: %30
- Ders Kitabı
 - Programlama Dilleri Prensipleri Ders Notları
 - Robert W. Sebesta, Concepts of Programming Languages, Pearson
- İletişim
 - Eposta: r.sinan.arслан@bozok.edu.tr



Ders İçeriği

- Giriş
- Dillerin Tarihçesi ve Çeşitleri
- Dillerin Tanımlanması ve Tasarımı
- Dillerin Çevrimi
- Temel Programlama Kavramları
- Veri Tipleri ve Yapıları
- Bağlama Kavramı ve İsim Kapsamları
- Yapısal Programlama
- Alt Programlar ve Modülasyon
- Nesneye Dayalı Programlama
- İstisnai Durumlar ve Önlemleri
- Eş Zamanlılık ve Thread Yapısı
- Fonksiyonel Programlama
- Mantıksal Programlama



- Genel Kavramlar
- Programlama Dili Nedir?
- Programlama Dillerinin sınıflandırılması
- Programlama Dillerinin Değerlendirme Ölçütleri
- Dil Seçimi Etkileyen Etkenler
- C Programlama Dili
- Java Programlama Dili
- Komut Satırı Parametreleri
- Lisp Programlama Dili
- Prolog
- Sorular



Genel Kavramlar

- Bilgisayarı oluşturan fiziksel parçaların tümüne donanım denilmektedir.(anakart, hafıza birimleri, giriş birimleri vb.)
- Kullanıcı ile Donanım arasındaki iletişimi sağlayan programlama ve programlama ile ilgili konuların geneline yazılım denilmektedir.
- Yazılım
 - Programlama Dilleri
 - Bu diller kullanılarak yazılmış kaynak programlar
 - çeşitli amaçlarla oluşturulmuş dosyalardan meydana gelirler.



Genel Kavramlar

- Yazılımlar Uygulama Alanlarına göre 5 gruba ayrılabilirler.
 - Bilimsel ve Mühendislik Yazılımları(Scientific and Engineering Software)
 - Bilimsel ve mühendislik konularındaki problemlerin çözülmesinde kullanılan programlardır.
 - Bu tür programlar hesaplamalı ve istatistiksel algoritmalar yoğun olarak kullanılırlar.
 - Ağırlıklı olarak CPU(Merkezi İşlem Birimi)'ni kullanırlar.
 - Elektronik devre çözümü yapan programlar, istatistik analiz programları bu türe örnek verilebilir.



Genel Kavramlar

- Yazılımlar Uygulama Alanlarına göre 5 gruba ayrılabilirler.
 - Mesleki Yazılımlar(Business software)
 - Veri oluşturma, veri işleme, veri saklanması ve verilere ulaşma ile ilgili özel amaçlı yazılımlardır.
 - Veri tabanı yazılımları, Stok kontrol/takip yazılımları, müşteri takip programları, muhasebe programları örnek olarak verilebilir.
 - Yapay Zeka Yazılımları(Artificial Intelligence Software)
 - İnsan düşünme ve öğrenme biçimini taklit etmeyi amaçlayan yazılımlardır.
 - Akıllı sistemler, robot yazılımları örnek olarak verilebilir.



Genel Kavramlar

- Yazılımlar Uygulama Alanlarına göre 5 gruba ayrılabilirler.
 - Görüntüsel Yazılımlar
 - Ağırlıklı olarak bilgisayarın grafik arabirimini kullanan, görüntü işleme ağırlıklı programlardır.
 - Oyun ve Animasyon yazılımları örnek olarak verilebilir.
 - Sistem Yazılımları(System Software)
 - Bilgisayar donanımsal kısmını yöneten yazılımlardır.
 - Yorumlayıcılar, derleyiciler, haberleşme programları, işletim sistemi ve yazı editörleri örnek olarak verilebilir.



Programlama Dili Nedir?

- Programlama Dili Nedir?

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

Programlama dillerinde o programlama dili için hem insanlar hem de bilgisayarlar tarafından ortak anlaşılacak kurallar ve semboller bulunmaktadır.

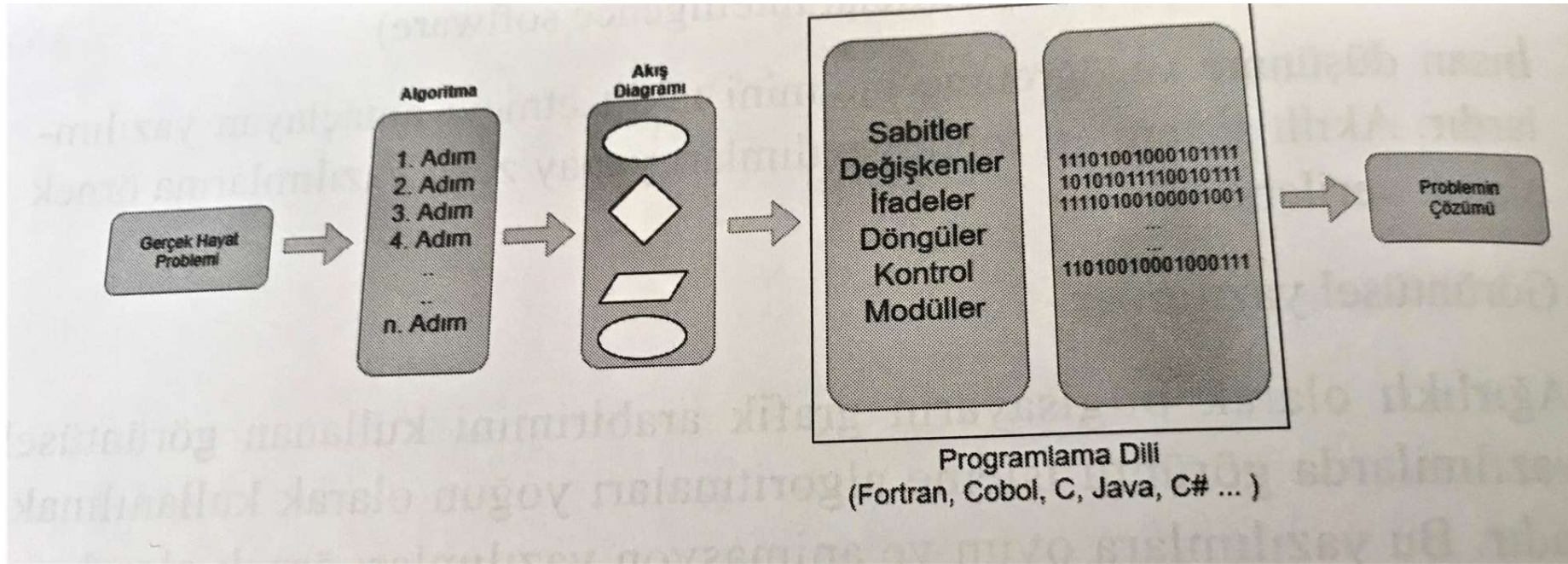


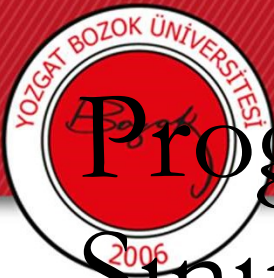
Programlama Dili Nedir?

- Bir Programlama Dili aşağıdaki amaçları karşılamalıdır
 - Bir programlama dili makinalara talimat vermek için gerekli bir araçtır.
 - Programcılar arasında iletişim için gereklidir.
 - Yüksek seviyeli tasarımları ifade etmek için gereklidir.
 - Algoritmaları çözmeye yarayan bir notasyondur.
 - Genel kavramlar arasındaki yakınları ifade etmeye yarayan bir yoldur.
 - Çözümlerin ve çözüm yollarının test edilmesi için gerekli bir araçtır.
 - Bilgisayarlı cihazları kontrol etmek için gerekli bir vasıtaadır.

Programlama Dili Nedir?

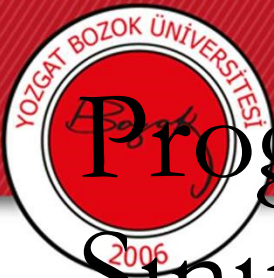
- Bir Programlama dilinin problemin çözümündeki yeri





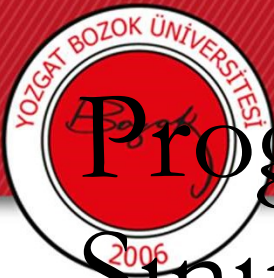
Programlama Dillerinin Sınıflandırılması

- Programlama Dilleri farklı şekillerde sınıflandırılabilir. Ancak en temel metoda göre aşağıdaki şekildedir.
 - Seviyelerine göre sınıflandırma
 - Uygulama alanlarına göre sınıflandırma
 - Dillerin tasarım paradigmalarına göre sınıflandırma



Programlama Dillerinin Sınıflandırılması

- Programlama Dillerinin Seviyelerine göre sınıflandırılması
 - Programlama dilleri, seviyelerine göre düşük ve yüksek düzeyli olarak ikiye ayrılırlar.
 - Bir programlama dilinin seviyesi kavramından o programlama dilinin bilgisayar donanımından soyutlanma derecesi ve insan algısına yakınlık derecesi anlaşılmalıdır.
 - Bir programlama dilinin seviyesi, insan algılamasına yaklaştıkça artar. Dolayısı ile ne kadar bilgisayar donanımına yakınsa seviyesi o kadar düşer.
 - Programlama dilinin seviyesi yükseldikçe bu dilin öğrenilmesi ve çalışması(yazılım üretmesi) kolaylaşır.



Programlama Dillerinin Sınıflandırılması

- Makine Dili
 - Bir bilgisayarın doğrudan anlayabildiği ve bilgisayarın donanım tasarımına bağlı bir dildir.
 - Herhangi bir programlama dili ile yazılan programın bilgisayar tarafından anlaşılabilmesi için makine diline çevrilmesi gereklidir.
 - Makine dilinde yazılan kodlar doğrudan makinenin işlemcisine, donanım parçasına verilen komutları içerir.
 - Farklı donanım parçaları kullanıldığında kod çalışmayacaktır. Tekrar yeniden yazılması gereklidir. Bir makine kodu yalnızca bir CPU versiyonuna uygulanabilir.



Programlama Dillerinin Sınıflandırılması

- Makine Dili
 - Makine dili ile program geliştirmek zor ve zahmetli bir süreçtir.
 - Sürekli güncellenmesi gereklidir.
 - Daha önce yazılmış bir makine kodunu okuma ve anlamakta oldukça zordur.
 - 1950'li yıllarda bu dilin sorunlarını çözmek için kısmen daha üst bir seviye dil olan Assembly languages(makine dilleri) geliştirildi.
 - Bu diller sayesinde 1 veya 0'lar yerine ingilizce bir takım kısaltmalar geliştirilmiştir. Ancak bu kısaltmalar kullanılarak yazılan kodlar öncelikle makine diline çevrilmekte sonrasında kullanılmaktadır. Bu daha yazılan kodun çok yavaşlamasına sebep olmaktadır.

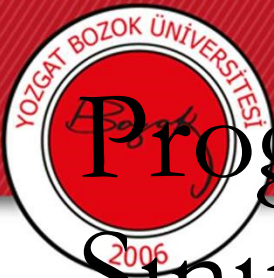


Programlama Dillerinin Sınıflandırılması

- Makine Dili(assembly)

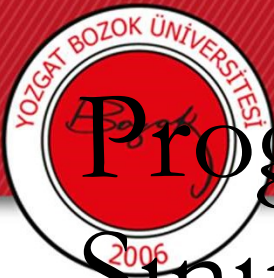
```
mesaj db 'Merhaba', 0x0d, 0x0a, '$'  
mov dx, mesaj  
mov ah, 9  
int 0x21
```

ekrana Merhaba yazma kodu



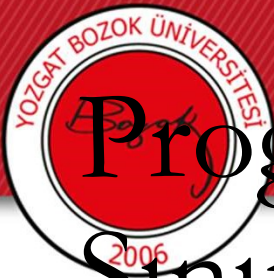
Programlama Dillerinin Sınıflandırılması

- Makine Dili
 - Aşağıdaki 2 yorumlayıcı ilk dönemde ortaya çıkan ve Assembly dillerini yorumlayan en önemli yorumlayıcı sistemleridir.
 - UNIVAC 1 (1950) – John Mauchly
 - IBM 701 (1953) – Speechcoding- John Backus
 - Bu yorumlayıcılar makine koduna çevirme işini Assembly kodlarını belleğe yüklendikten sonra yaptığı için çok yavaş çalışıyorlardı.
 - Bu sebeple, Grace Hopper tarafından yorumlayıcı yerine «compiler» derleyici fikri ortaya atıldı.



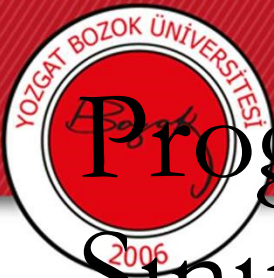
Programlama Dillerinin Sınıflandırılması

- Makine Dili
 - Derleyici sayesinde program kodlarının bir defa makine koduna çevrilmesi böylece sonrasında tekrar tekrar kullanılabilir olması sağlanmak istenmiştir.
 - Böylece Assembly programları derleyici sayesinde makine koduna çevrilmeye başlanmıştır.
 - Assembly dili ile geliştirme süreçlerinde bu dil ile gelişmiş yazılımların oluşturulması oldukça uzun sürmekte idi. Bu sebeple daha yüksek seviyeleri dillerin geliştirilmesi fikri ortaya atılmıştır.
 - Bu zamana kadar yüzlerce farklı algoritmik temelli yüksek seviye diller üretilmiştir.



Programlama Dillerinin Sınıflandırılması

- ÜST SEVİYELİ DİLLER
 - FORTRAN(FORmula TRANslator): Kompleks matematiksel hesaplamalar gerektiren mühendislik ve bilimsel uygulamalardan kullanılmak üzere 1954-1957 yılları arasında IBM - John Backus tarafından geliştirilmiştir. Halen matematiksel problem çözümlerinde kullanılmaktadır. FORTRAN IV ve 77 olmak üzere 2 temel versiyonu vardır. İlk üst düzey dil olarak kabul edilmektedir.



Programlama Dillerinin Sınıflandırılması

- ÜST SEVİYELİ DİLLER

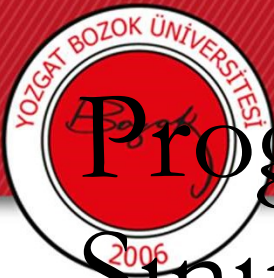
- COBOL(COMmon Business Oriented Language): 1959 yılında, Amerikan bilgisayar üreticileri, özel ve devlet sektöründen seçilmiş bir grup tarafından geliştirilmiştir. Veri yönetimi ve işletmelerin ticari faaliyetlerinin yürütülmesi amacıyla kullanmak üzere geliştirilmiştir.
- ALGOL(The ALGORitmic Language): 1958 yılında Avrupada bir ekip tarafından geliştirilmiştir. FORTRAN diline alternatif olarak önerilmiştir.



Programlama Dillerinin Sınıflandırılması

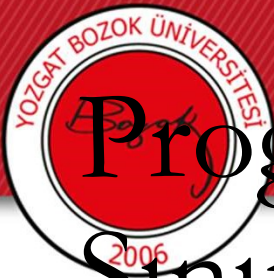
- ÜST SEVİYELİ YAPISAL DİLLER

- PASCAL: Yapısal programlama kavramı üzerine odaklanılmıştır. 1971 yılında Niclaus Wirth tarafından geliştirilmiştir. Eğitim amaçlı bir dildir. Ticari ve endüstriyel uygulamalarda çok yaygın kullanılmamıştır. Modula ve Modula-2 diller baz alınarak hazırlanmıştır.
- BASIC: 1960 yılların ortalarında ortaya çıkmıştır. Yüksek seviyeli dillerin eskisi olup, öğrenmesi en kolay ve basit olanlardan birisidir. Ticari uygulamalarda yaygın olarak kullanılmıştır. ANSI tarafından standartlaştırılmıştır. Microsoft tarafından geliştirilen Visual BASIC programı ile nesne yönelimli bir çok özellik bulunmaktadır.



Programlama Dillerinin Sınıflandırılması

- ÜST SEVİYELİ YAPISAL DİLLER
 - ADA: Amerikan savunma departmanı desteği ile 80'li yılların başında geliştirilmiştir. Genel amaçlı bir dildir. Ticari uygulamalardan bilimsel uygulamalara kadar geniş bir alanda kullanılmaktadırlar. Gerçek zamanlı uygulama geliştirme desteği vermektedir. Büyük derleyicilere ihtiyaç duyması sebebiyle C, C++ dillerine karşı geri planda kalmıştır.
 - JAVA, C#: 4. kuşak dillerdir. algoritmik yapı içermeyen programların görsel bir ortamda hazırlanabildiği dillerdendir. 4GL olarak kısaltılırlar.



Programlama Dillerinin Sınıflandırılması

- ÜST SEVİYELİ YAPISAL DİLLER ÖZET
 - Çok yüksek seviyeli diller yada görsel diller
 - Access, Xbase, Visual Basic, Oracle forms vb.
 - Yüksek seviyeli diller, algoritmik diller
 - Fortran, pascal, basic, cobol
 - Orta seviyeli diller
 - Ada, C
 - Alçak seviyeli diller
 - Assembly
 - Makine dili
 - 1 ve 0'lardan oluşan ilkel seviye programlama dili.



Programlama Dillerinin Sınıflandırılması

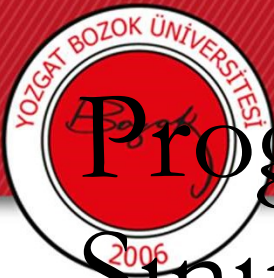
- ÜST SEVİYELİ YAPISAL DİLLER ÖZET

Dil	Seviye	Programlama Kolaylığı	Görelî Hızı	Gücü ve Esnekliği
Java	Yüksek	Kısmî Kolay	Yavaş	Zayıf
Basic	Yüksek	Kolay	Yavaş	Zayıf
Cobol	Yüksek	Kolay	Yavaş	Zayıf
Pascal	Yüksek	Kısmî Kolay	Yavaş	Zayıf
Fortran	Yüksek	Orta	Orta	Orta
C	Orta	Kısmî Zor	Hızlı	İyi
Assembly	Düşük	Zor	Çok Hızlı	Çok İyi



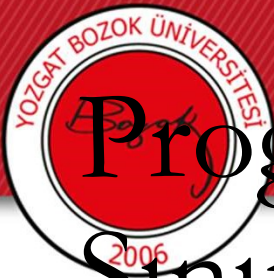
Programlama Dillerinin Sınıflandırılması

- PROGRAMLAMA DİLLERİNİN UYGULAMA ALANLARINA GÖRE SINIFLANDIRMA
 - Bilimsel ve mühendislik uygulama dilleri: Pascal, C, FORTRAN, geliştirilen ilk diller bu alanda olmuştur.
 - Veritabanı dilleri: Veri ekleme silme güncelleme için hazır komutlar içerir. XBASE, Oracle , MSSQL, Visual Foxpro.
 - Genel amaçlı programlama dilleri: Bir çok kesimin kullanımına olanak sağlar. Günümüz popüler uygulamaları bu alandadır. C, C#, Java, Basic
 - Yapay zeka dilleri: İnsan düşünme ve karar verme biçimini modeller. Prologi Lisp.
 - Modelleme, Simülasyon dilleri: Simula 67, Elektronik Workbench, GPSS, gerçek hayat problemlerini koda yansıtacak araçlar içerirler.



Programlama Dillerinin Sınıflandırılması

- PROGRAMLAMA DİLLERİNİN UYGULAMA ALANLARINA GÖRE SINIFLANDIRMA
 - Makro dilleri: Ayrıntılı dil ifade dizileri oluştururlar. awk, Perl, Python, Tcl, JavaScript, Excel.
 - Sistem Programlama dilleri: Bilgisayar donanımı ile arabirim oluşturan, uygulama programlarına çeşitli yönlerden hizmet veren programlardır. C dili en tanınmış örnektir. BCPL, C++'da bu diller arasından incelenir. UNIX gibi bir çok işletim sisteminin çekirdeği C dili kullanılarak yazılmıştır. Oldukça güçlü bir dildir.
 - Ticari uygulamalara yönelik diller: Veri işleme, sayısal hesaplar yapma için kullanılırlar. COBOL örnek verilebilir.



Programlama Dillerinin Sınıflandırılması

- DİLLERİN TASARIM PARADİGMALARINA GÖRE SINIFLANDIRMA

Paradigma: bir grubun ortak kararlarını temsil eden ve grubun konuya yeni bir şekilde bakışını sağlayan bir kavramsal şemadır. Farklı paradigmalar, farklı çözüm yaklaşımına ve farklı programlama stilini gerektirir.

- Emir esaslı(imperative)
- Nesneye yönelik(object oriented)
- Fonksiyonel(functional)
- Mantık esaslı(logical)



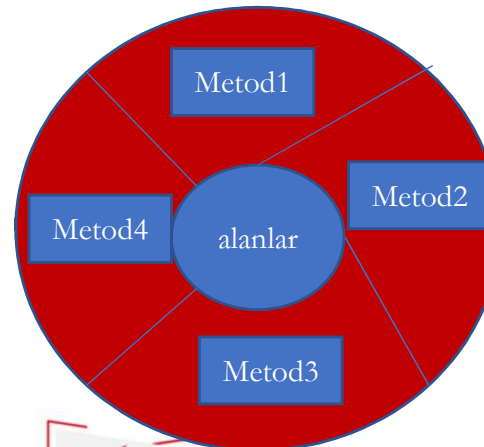
Programlama Dillerinin Sınıflandırılması

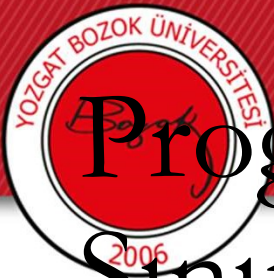
- DİLLERİN TASARIM PARADİGMALARINA GÖRE SINIFLANDIRMA
 - Emir esaslı(imperative): Bu diller işlem tabanlı olup, bir program bir dizi işlem olarak görülür. programlama deyimleri, birbirleri ile değişkenler aracılığıyla iletişim kurar. bir program bir dizi deyim içerir ve her bir deyim bellekte bir veya birden fazla bellek alanının değişmesine neden olur.
 - C, FORTRAN, Pascal, COBOL, ADA gibi diller tarafından desteklenir.

```
C dili  
x= 10  
Pascal  
x:=10  
APL  
x <-- 13  
Lisp  
(setq x 10)
```

Programlama Dillerinin Sınıflandırılması

- DİLLERİN TASARIM PARADİGMALARINA GÖRE SINIFLANDIRMA
 - Nesneye yönelik(object oriented): SIMULA 67 dili temel alınarak oluşturulmuştur. Nesnelerin sınıf ve alt sınıflara gruplanması paradigmasını oluşturur. Karmaşık veri nesneleri ve bu nesneler üzerinde oluşturulacak metodlar tasarlanır. Daha basit nesnelerin toplanması ile karmaşık nesnelere ulaşılır. C++, Java, C# örnek verilebilir.





Programlama Dillerinin Sınıflandırılması

- DİLLERİN TASARIM PARADİGMALARINA GÖRE SINIFLANDIRMA
 - Fonksiyonel Diller: Veriler ve sonucu elde etmek için veriye uygulanacak fonksiyonel dönüşümler bu paradigmanın temelini oluşturur. LİSP, Scheme, ML dilleri örnek olarak verilebilir.

```
Function n( ... (Function2(Function1(Data))) ...)
  (Defun kontrol(yas)
    (
      (if
        (and (> yas 2) (< yas 18) T
          nil
        )
      )
    )
  )
```

Lisp dili kod örneği



Programlama Dillerinin Sınıflandırılması

- DİLLERİN TASARIM PARADİGMALARINA GÖRE SINIFLANDIRMA
 - Mantıksal Diller: Bir işin nasıl yapılacağıının belirtilmesi yerine, ne yapılması istendiğinin belirtilmesi olarak görülür. sürekli bir koşula bağlılık esastır. Prolog dili bu paradigmanın benimsendiği en bilinen dildir. mantıksal diller imperative diller deyimsel ifadeler içerirler ancak sıralı olmak zorunda değildir.
 - Prolog, insan makine etkileşimine uygundur. Özellikle bir mantık yürütme şeklini kullanarak kendisine yöneltilen sorgulara yanıt verir.

```
Şart_1 -> Komut_1  
Şart_2 -> Komut_2  
Şart_3 -> Komut_3  
Şart_4 -> Komut_4
```

Merhaba:-

```
nl, // newline  
write ('Merhaba')
```



Programlama Dillerinin Değerlendirme Ölçütleri

- Her programlama dilinin farklı bir düşünce sistemi vardır. Çözülecek problemin tipine ve uygulama alanlarına göre programlama dilleri arasında seçim yapmak için değerlendirme ölçütlerine ihtiyaç duyulmaktadır. Bir programlama dili ile çalışmaya karar vermeden bu ölçütlere göre değerlendirmek en iyi yöntem olacaktır.



Programlama Dillerinin Değerlendirme Ölçütleri

- Bu ölçütlerden bazıları:
 - Giriş/Çıkış Birimi Kolaylığı
 - Taşınabilirlik
 - Alt Programlama Yeteneği
 - Verimlilik
 - Okunabilirlik, Yazılabilirlik
 - Esneklik
 - Öğrenme Kolaylığı
 - Genel Amaçlılık
 - Yapısalılık
 - Nesne Yönelimlilik



Programlama Dillerinin Değerlendirme Ölçütleri

- Bu ölçütlerden bazıları:
 - İfade Gücü
 - Veri Türleri Ve Yapıları
 - Giriş/Çıkış Birimi Kolaylığı
 - Taşınabilirlik
 - Alt Programlama Yeteneği
 - Verimlilik
 - Okunabilirlik, Yazılabilirlik
 - Esneklik
 - Öğrenme Kolaylığı
 - Genel Amaçlılık
 - Yapısalılık
 - Nesne Yönelimlilik



Programlama Dillerinin Değerlendirme Ölçütleri

- Bu ölçütlerden bazıları:
 - İfade Gücü(expression power): Algoritmayı tasarlayan kişinin çözüm düşüncesini ve niyetini açık bir biçimde yansıtabilmesine olanak tanıyan dil istenir. C, PASCAL gibi diller ifade gücü yüksek dillerdendir.
 - Veri türleri ve yapıları(data types and structures): çeşitli veri türlerini(tamsayı, gerçek sayı, karakter vb.), ve veri yapılarını(diziler, bağlı liste, kuyruk yapıları vb.) destekleme yeteneğini ifade eder. C, PASCAL, JAVA dilleri veri yapıları bakımında zengin dillerdendir.



Programlama Dillerinin Değerlendirme Ölçütleri

- Bu ölçütlerden bazıları:
 - Giriş/Çıkış Birimi (I/O) Kolaylığı: Programlama yazmayı kolaylaştıran ve ifade gücünü artıran bir özelliktir. Veri tabanı programları(DBASE, PARADOX) bu yetenekler konusunda oldukça üstündürler. Sıralı, indeksli, rasgele dosyalara erişme, veri tabanı kayıtlarını geri alma, güncelleştirme ve sorgulama konusunda C dili benzer diller yetkin değildir. Hepsinde özel kütüphanelere ihtiyaç duyulur.
 - Taşınabilirlik(portability): Bir programlama dilinde yazılmış kaynak kodun başka sistemlerde de sorunsuz derlenerek çalıştırılabilmesini ifade eder. Dillerin seviyesi düştükçe taşınabilirlik azalır. Taşınabilirliğin sağlanması için standartlaştırma gereklidir. ISO(International Standard Organization) ve ANSI (American National Standart Institute) tarafından programlama dilleri standardize edilirler. C dili 1989 yılında standardizasyon işlemlerinin tamamlanması sebebiyle daha taşınabilir bir dildir.



Programlama Dillerinin Değerlendirme Ölçütleri

- Bu ölçütlerden bazıları:
 - Alt programlama yeteneği(Modularity): Bir programlama dilinin, programı parçalar halinde yazmayı desteklemesidir. Karmaşık problemlerin çözümünde istenen bir özelliktir. Çok tekrarlanan programların alt programlar haline getirilmesine imkan tanınması yapısal programlama dillerinin vazgeçilmez unsurudur. Alt programlama sayesinde, programların anlaşılması kolaylaşır, kodun okunabilirliği artar, kaynak kodu bakımı ve yeni kullanılması olanağı artar. C alt programlama yeteneği oldukça gelişmiş bir dildir. fonksiyon sayesinde çalışılması onu yetenekli hale getirir.



Programlama Dillerinin Değerlendirme Ölçütleri

- Bu ölçütlerden bazıları:
 - Verimlilik(Efficiency): Bir dilde yazıldıktan sonra amaç koduna dönüştürülmüş programların hızlı çalışabilmesine verimlilik denir. Derleyici, dil seviyesi ve dilin genel yapısı, algoritma, işlemci ve RAM bellek gibi birçok faktörden etkilenir. C programları hızlı çalışır ve az yer kaplarlar.
 - Okunabilirlik(Readability): kaynak kodun çabuk ve kolay bir biçimde algılanabilmesi anlamına gelir. Bu sayede güncelleştirmeyi kolaylaştırır ve bir çok kişinin ortak kodlar üzerinde çalışmasına imkan tanır. En kolay okunabilir kod en iyi kod olarak kabul edilir.



Programlama Dillerinin Değerlendirme Ölçütleri

- Bu ölçütlerden bazıları:
 - Esneklik(Flexibility): Programlama dilinin programcayı kısıtlamamasını ifade eder. Bu durum deneyimsiz programcılar için sorun iken, deneyimliler için oldukça iyi bir özelliktir. C dili bu esnekliği sağlayabilen dillerden birisidir.

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

- Örnek bir c kodu. karakterden int değere veya tersi şekilde çeşitli atamalar yapmaya imkan tanır. bu onun esneklik özelliğinden gelir.



Programlama Dillerinin Değerlendirme Ölçütleri

- Bu ölçütlerden bazıları:
 - Öğrenme kolaylığı(pedagogy): bir programlama dili donanımdan uzaklaştıkça yani seviyesi yükseldikçe öğrenme ve uygulama geliştirme kolaylaşır. yüksek seviyeli dillerin popüler olmasının sebebidir. Ancak bazı üst seviye dilleri, C# gibi, öğrenmek kolay değildir ve bazı ön şartlar gerektirebilir.
 - Genel amaçlılık(generality): programlama dillerinin çok çeşitli uygulamalarda etkin olarak kullanılabilmesidir. COBOL ticari uygulamalarda etkin bir dil iken mühendislik uygulamalarında tercih edilmez. C, Java gibi diller genel amaçlı diller olarak gösterilebilir.



Programlama Dillerinin Değerlendirme Ölçütleri

- Bu ölçütlerden bazıları:
 - Yapısallık: Bir programlama tekniğidir. 1960 yılında geliştirilmiştir. bütün programlama dilleri bu tekniği benimseyecek şekilde geliştirilmişlerdir. blok halinde yazım ve alt programlama yoğun olarak kullanılır. soyutlama mümkündür. 4 temel ilkesi vardır.
 - böl ve yönet
 - veri gizleme
 - tek giriş ve tek çıkış
 - döngüler ve diğer kontrol yapıları



Programlama Dillerinin Değerlendirme Ölçütleri

- Bu ölçütlerden bazıları:
 - Nesne Yönelimlilik (Object Oriented): Veri + program = nesne yapısına uygun olarak çalışan ve programların nesnelerle yapıldığı dillerdir.
 - kaynak kodların çok büyümesi yüzünden 1980'li yıllarda ortaya çıkan bir programlama tekniğidir.
 - C dilinin ortaya çıktığı yıllarda büyük ölçekli programlar yapılmadığı için nesne tabanlı geliştirme kütüphaneler bulunmamaktadır. Sonrasında bu eklemeler yapılarak C++ dili geliştirilmiştir. Pascal diline eklemeler yapılarak Delphi, Cobol diline eklemeler yapılarak OOCobol ve Ada diline eklemeler yapılarak Ada 95 dilleri ortaya çıkmıştır.



Programlama Dillerinin Değerlendirme Ölçütleri

- Bu ölçütlerden bazıları:
 - Nesne Yönelimlilik (Object Oriented):
 - Java gibi bazı programlama dilleri ise doğrudan nesne tabanlı programlama tekniğini destekleyecek şekilde tasarlanmıştır. c# ise tamamen nesne yönelimli bir dildir.



Dil Seçimini Etkileyen Etkenler

Karakteristik	Okunabilirlik	Yazılabilirlik	Güvenirlik
	X	X	X
Sadelik (Simplicity)	X	X	X
Kontrol Yapısı	X	X	X
Veri Tipi ve Yapısı	X	X	X
Sözdizim Tasarımı (Syntax)		X	X
Soyutlama Desteği		X	X
İfade Gücü			X
Tip Kontrolü (Type Checking)			X
Hata Ayıklama (Exception Handling)			X
Örtüşmeleri Sınırlama (Restricted Aliasing)			X



C Programlama Dili

- Dennis Ritchie tarafından 1972 de Bell Laboratuvarlarında geliştirilmiştir.
- Taşınabilir ve hızlı bir dildir.
- imperative prosedürel bir dildir. Sınıf kavramını desteklemez.
- Unix gibi işletim sistemleri C dili ile geliştirilmiştir.
- C dili kullanılarak bilgisayar donanımı kontrol edilebilir ve bellekte bitler üzerinde işlem yapılabilir.
- Esnek bir dildir. Bu sebeple birçok programlama dilinde yapılmayacak hatalar bu dilde yapılabilir.



C Derleyicisi Kurulumu ve Örnek kodlar ile çalışmalar

- MinGW GNU C derleyicisinin kurulumu
- Örnek bir C kodunun Derlenmesi
- Make Dosyası



C Derleyicisi Kurulumu ve Örnek kodlar ile çalışmalar

- Program kaynak kodları klasör hiyerarşisi
 - bin
 - include
 - lib
 - src
 - makefile



C Derleyicisi Kurulumu ve Örnek kodlar ile çalışmalar

- Örnek C kodunu derlemek için makefile dosyası örneği

DERLE

```
gcc -I ./include -o ç/lib/Kisi.o -c ./src/Kisi.c
```

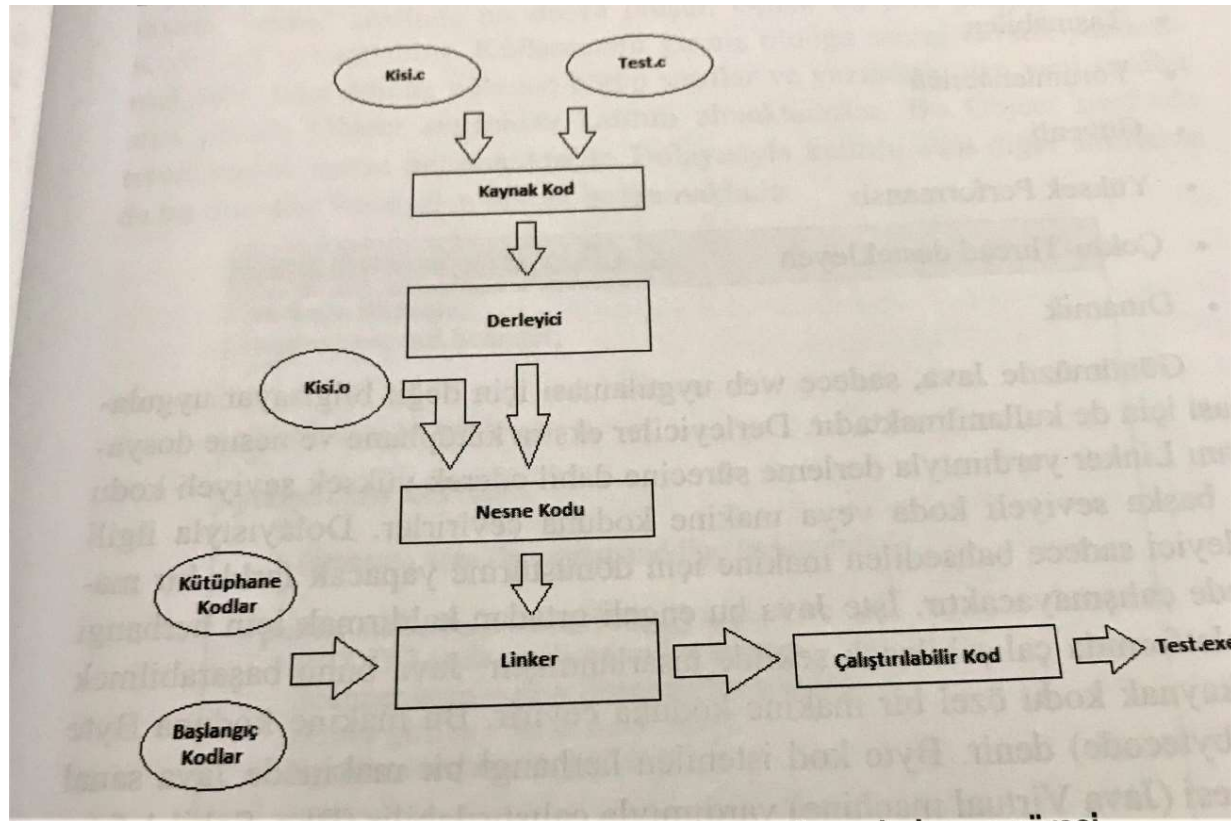
```
gcc -I ./include -o ./bin/Test ./lib/Kisi.o ./src/Test.c
```

CALIŞTIR

```
./bin/Test
```


C Derleyicisi Kurulumu ve Örnek kodlar ile çalışmalar

- C programlama dilinde kodların derlenme süreci,





Java Programlama Dili

- İlk olarak Oak ismi ile tasarlanmış olup gömülü uygulama geliştirmek için önerilmiştir.
- Daha sonra, internet uygulamaları geliştirmek için 1995 yılında Java ismi ile yeniden tasarlanmıştır.
- «Kodu bir kere yaz istediğin yerde çalıştır» sözü sayesinde hızlı bir şekilde popüler olmuştur.
- Java dilinin temel özellikleri
 - Tasarımı Kolay
 - Nesne Yönelimli
 - Taşınabilen
 - Yorumlanabilen
 - Güvenli
 - Yüksek Performanslı
 - Çoklu-thread Destekleyen
 - Dinamik



Java Programlama Dili

- Günümüzde web uygulamaları yanında bir çok bilgisayar uygulamasını geliştirmek içinde kullanılmaktadır.
- Derleyiciler derleme sürecinde farklı kütüphane ve nesne dosyasını da program dosyalarına eklerler ve makine koduna dönüştürürler. Dolayısı ile makine koduna dönüştürülen program farklı bilgisayarda çalışmaz. JAva bu engeli ortadan kaldırmak için bir platforma tasarlamıştır.
- Java bunu kaynak kodu özel bir makine koduna dönüştürür(Byte code). Bu byte kod istenilen makineye atılır ve Java Virtual Machine(JVM) sayesinde çalıştırılır.



Java Programlama Dili

- Applet ve Servlet olarak yazılabilirler.
- Her java programı en aza bir sınıfa sahiptir. Sınıf metod ve verileri içerir. «.java» uzantılıdır. nesne dosyası oluşturulduğunda «.class» şekline dönüşür.
- Örnek bir Java dosyasının yazılması ve incelenmesi



Lisp Programlama Dili

- Fonksiyon programlama bir fonksiyon aynı parametreleri aldığı sürece aynı sonucu üretir.
- Daha çok yapay zeka için kullanılan bu diller, benzetim(simulasyon) uygulamalarında da kullanılmaktadır.
- Sadece fonksiyonlar üzerine kurulmuş modelleri gerçekleştirebilirler.
- Fonksiyonlar birden çok parametre alabilirler ama geriye sadece 1 parametre döndürebilirler.
- Fonksiyonların, recursive şekilde kendi kendini çağırması mümkündür.



Lisp Programlama Dili

- Lisp, sembolik veri işleme amacı ile tasarlanmıştır.
- Karmaşık hesaplamaları daha basit ifadeler cinsinden yazarak kolaylıkla çözümlenmesini sağlar.
- Atom ve liste isimli iki veri yapısı içerir.
- Atom semboller ve sayısal değerleri alır.
- Yorumlayıcı bir dildir. Ancak derleyici kullanan sürümleri de mevcuttur. GNU Common Lisp.

```
(Defun Merhaba()  
  «Merhaba»  
)
```

Örnek Listp Kodu



Lisp Programlama Dili

- GNU Common Lisp kurulumu



Prolog Programlama Dili

- Prolog dili 1970 yılında geliştirilmiştir.
- Mantıksal bir programlama dilidir. Bildirme esaslı bir yapısı vardır.
- Bu yapıda 2 temel konsept vardır. Olaylar ve kurallar.
- Doğru olan durumlar olayları oluşturur. Bu durumda Prolog dili bir durumun doğru olduğunu ispatlamak için kullanılabilir.
- Lisp gibi daha çok yapay zeka uygulamalarında faydalanılır.
- Kuralları ifade etmek için önermelerden yararlanır.
- Atomik önerme ve bileşik önerme türleri vardır.
- Fonksiyonel dillerin bir çok özelliğini sağlar. Örnek: özyinelemeli çağırım.



- Assembly ile Makine dili arasındaki farklar nelerdir?
- C dilinde başlık dosyası kullanmanın amacı nedir?
- Lisp dili yorumlayıcıya mı yoksa derleyiciye mi sahiptir?