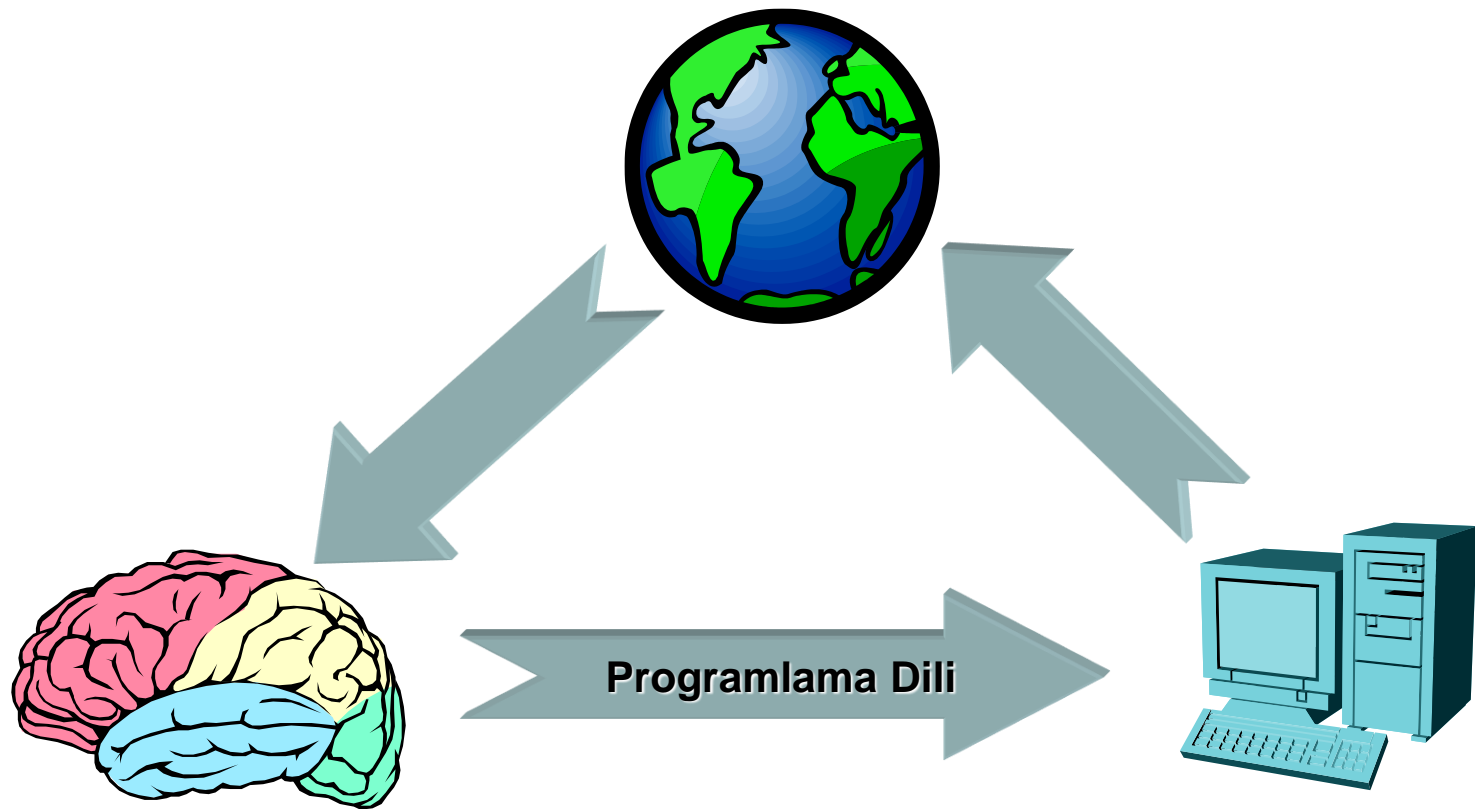


Bölüm Hedefi

- Programlama dillerinin tanımı,
- Dillerin kuşaklara ayrılması
- Programlama dillerinin sınıflandırılması,
- Programlama dilleri değerlendirilmesinde kullanılan ölçütler,
- Temel programlama paradigmaları

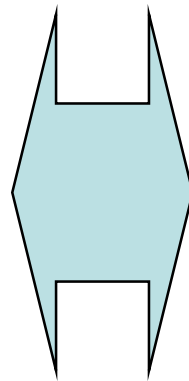
PROGRAMLAMA DİLİ NEDİR?

- **Programlama dili**, bir problemin çözümünün bilgisayardaki gerçekleştirimini ifade etmek amacıyla tasarlanmış ve o programlama dili için **hem insanlar hem de bilgisayarlar tarafından ortak olarak anlaşılacak** kurallar ve semboller dizisidir.



Programlama

```
int sum(int[] x) {  
    int sum = 0;  
    n = 0;  
    while (n < x.length) {  
        sum += x[n];  
    }  
    return sum;  
}
```



```
0010101010101010  
10101011111010  
11101010101110  
00101010101010  
...
```

```
program gcd(input, output);  
var i, j: integer;  
begin  
    read(i, j);  
    while i <> j do  
        if i > j then i := i - j;  
        else j := j - i;  
    writeln(i)  
end.
```



Compilation

```
27bdfdd0 afbf0014 0c1002a8 00000000 0c1002a8 afa2001c 8fa4001c  
00401825 10820008 0064082a 10200003 00000000 10000002 00832023  
00641823 1483fffa 0064082a 0c1002b2 00000000 8fbf0014 27bd0020  
03e00008 00001025
```

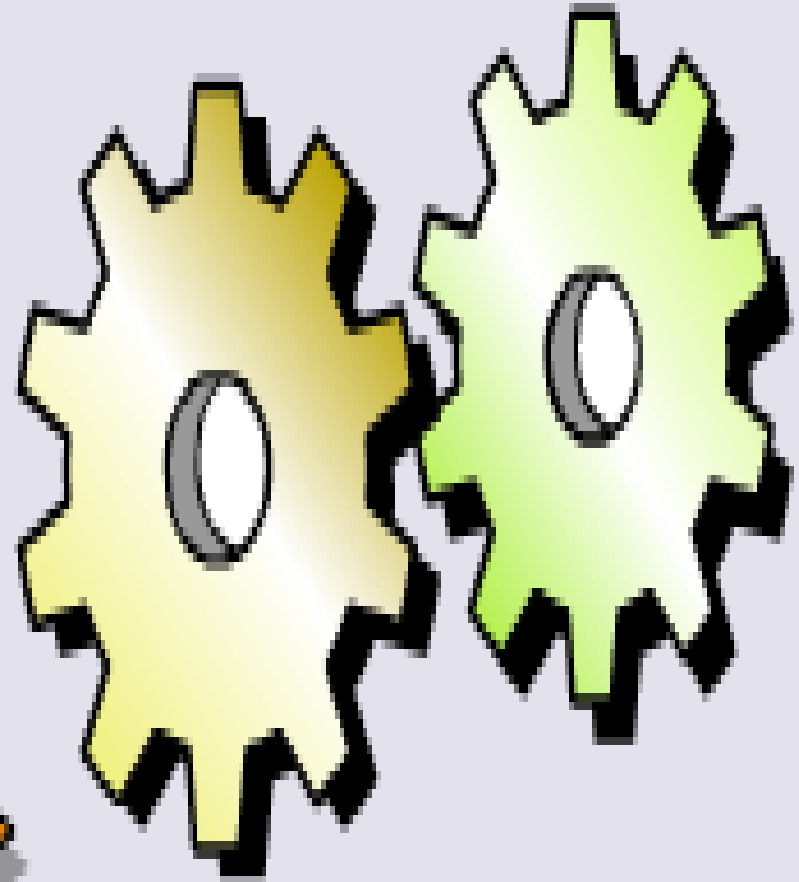
Makine Dili

- Bir programlama dilinin bilgisayar tarafından anlaşılması için, o dilin sözdiziminin ve anlamının makine diline çevrilmesi gereklidir.
- **Makine dili**, bir bilgisayarın doğrudan anladığı gösterim olup, bilgisayarların ana dili olarak nitelenebilir.

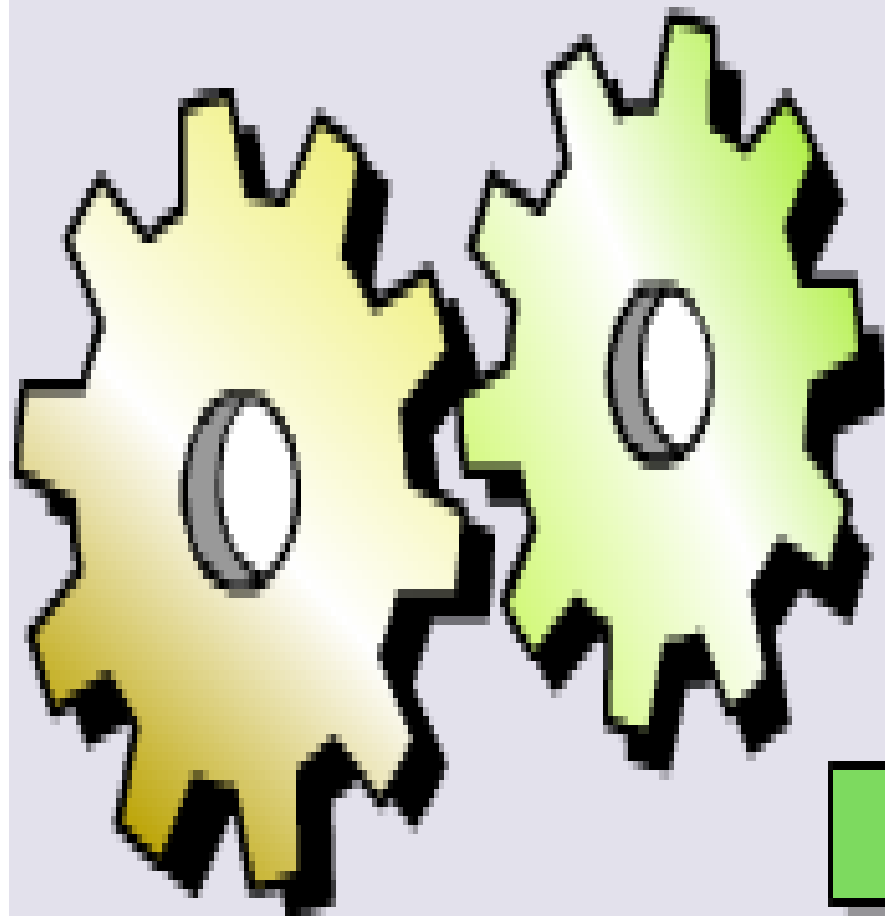


```
u function ()  
> new array [  
]; math ()  
xy_stop ();  
display (j < 0  
null) if abc;
```

PROGRAMLAMA DİLİ



DERLEME



DERLEME

```
00100101  
10101010  
01101101  
10110101  
10101101  
01001110
```



MAKİNE DİLİ

Dillerin kuşaklara ayrılması

- Düşük düzeyli programlama dilleri
- Yüksek düzeyli programlama dilleri

Düşük düzeyli programlama dilleri

- Dilin özellikleri bilgisayar donanımına bağlıdır
- Makine dili ve Asembler dili
- Makine dili sadece 0 ve 1'lerden oluşur ama makinenin ana özelliklerini içerir.
- Assembly dilinde ise işlemler, değerler ve bellek yerlerinin yerini isimler ve bazı semboller almıştır. (ADD R1 24)

Yüksek düzeyli programlama dilleri

- Bu diller bilgisayar donanımına bağlı özellikler taşımaz
- Okunabilir bir gösterim vardır
- Donanımdan bağımsızdır.
- Program kütüphaneleri sağlanmaktadır.

PROGRAMLAMA DİLLERİNİN GELİŞİMİ

- Programlama dili tasarım ve gerçekleştirmeleri, 1950'li yıllarda tanıtılan ilk yüksek düzeyli diller olan FORTRAN, COBOL ve LISP'den beri sürekli olarak gelişmiştir.
- Günümüzde hızla değişen bilgisayar teknolojileri, yeni gereksinimleri ortaya çıkarmaktadır. Bunun sonucu olarak, gelecekte de yeni programlama dillerinin geliştirilmesi kaçınılmazdır.

Örnekler

- **FORTRAN**
- **COBOL**
- **ALGOL**
- **PASCAL**
- **BASIC**
- **ADA**
- **JAVA**
- **C. C++, C#**
- **PROLOG**

Programlama Dillerinin Sınıflandırılması (Seviyesine göre)

- Çok Yüksek Seviyeli Programlama Dilleri (İnsana en yakın)
VISUAL BASIC, Access....(Dekleratif Diller)
- Yüksek Seviyeli Programlama Dilleri (PASCAL, COBOL)
- Orta Seviyeli Programlama Dilleri (C, ADA)
- Alçak Seviyeli Programlama Dilleri (Sembolik Makine Dilleri)
- Makine Dilleri (Bilgisayara en yakın)

Dillerdeki seviye yükseldikçe programcının işi daha kolay hale gelirken genel olarak esneklik ve verimlilik azalmaktadır.

Tablo 1.1 Programlam dilleri

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

Programlama Dillerinin Sınıflandırılması

- Programlama dilleri, **her dil grubunu diğerlerinden ayırt eden özellik** olduğu kabul edilen bir özelliğe göre sınıflandırılırlar. Bu sınıflandırmalar birbirini dışlayan sınıflar oluşturmaz ve gruplar birbirleriyle çakışabilirler.

Uygulama Alanlarına Göre

- Sayısal Uygulamalar için programlama dilleri
- Ticari Uygulamalar için programlama dilleri
- Yapay Zeka Uygulamaları için programlama dilleri
- Sistem programlama için programlama dilleri

Sayısal Uygulamalara Yönelik Programlama Dilleri

- Bilgisayarların ilk olarak kullanıldıkları alan sayısal uygulamaların ağırlıklı olduğu bilimsel çalışmalar olmuştur. Bu nedenle ilk geliştirilen programlama dilleri, sayısal programlama özelliklerini vurgulamışlardır.

FORTRAN

ALGOL

PL/I

BASIC

APL

SIMULA 67

PASCAL

C

ADA

Ticari Uygulamalara Yönelik Programlama Dilleri

- Ticari uygulamalardaki veri işleme, sayısal hesaplamalardan sonra ilk olarak gelişen uygulama alanıdır.
- **COBOL** (COmmon Bussiness Oriented Language) :
- A.B.D. Savunma Bakanlığı'nın, birçok şirketle birlikte İngilizce'ye yakın ve ticari uygulamalara yönelik bir dilin geliştirilmesi çalışmalarını desteklemesi sonucu, 1959 yılında COBOL tanıtılmıştır.
- COBOL dili, özellikle yoğun miktarda veri işleme kolaylıkları sağlayan deyimleri ve yapıları nedeniyle ticari uygulamalar alanında yazılım geliştirmek için popüler olmuştur.
- COBOL, hiyerarşik veri yapıları gibi birçok yeni kavram içeren ve özellikle raporlama açısından çeşitli olanaklar sağlayan bir dildir. 1961 ve 1962'de yenilenen dil, 1968'de standartlaştırılmış ve 1984'de tekrar yenilenmiştir.

Yapay Zeka Uygulamaları İçin Programlama Dilleri

- **LISP :**
LISP 1950'li yılların sonunda, *liste işleme* amaçlı fonksiyonel bir dil olarak, John McCarthy tarafından IBM 704 bilgisayarları için geliştirilmiştir. LISP, diferansiyel ve integral hesaplamalarında, sayısal mantık ve yapay zekanın diğer alanlarında sembolik hesaplamalar için kullanılmıştır. Sonraki yıllarda, birçok kez yenilenen LISP'ten başka, Scheme (1975) ve ML (1988) de LISP'i izleyen yapay zeka alanındaki fonksiyonel dillere örnektir.
- **PROLOG:** Prolog, temel denetim yapısı sembolik mantık kavramlarına dayanan özel amaçlı bir dildir. 1972 yılında tanıtılmış olan Prolog'un temel uygulama alanı, doğal dil işlemedir. Günümüze kadar Prolog, veritabanlarından uzman sistemlere kadar çeşitli uygulama alanlarında kullanılmıştır.

Sistem Programlama Dilleri

Sistem programlama dillerine en tanınmış örnek C programlama dilidir. Sistem programlaması alanında etkinlik gereksinimi nedeniyle, birleştirici dilleri yaygın olarak kullanılmıştır. 1970'li yılların başında C dilinin geliştirilmesi ile sistem programlama alanında da diğer uygulama alanlarında olduğu gibi yüksek düzeyli programlama dillerinin kullanımı yaygınlaşmıştır.

Programlama Dillerinin Sınıflandırılması

(Uygulama Alanlarına göre)

- Bilimsel ve Mühendislik Dilleri: Fortran, PASCAL, C, C++
- Veritabanı Programlama Dilleri: DBASE, PARADOX, FOXPRO, SQL
- Yapay Zeka Dilleri: LISP, PROLOG
- Genel Amaçlı Diller: C, PASCAL...
- Sistem Programlama Dilleri: C, Sembolik Makine Dilleri

Dil Değerlendirme Ölçütleri

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

İFADE GÜCÜ

- Algoritmayı tasarlayan kişinin niyetlerini açık bir biçimde yansıtabilmesine olanak tanıyan bir dil, ifade gücü yüksek bir dildir.
- Bir matematikçi kendi alanındaki sembolleri kullanmak isteyebilir
- C ve PASCAL dillerinin ifade gücü yüksektir.

VERİ TÜRLERİ VE YAPILARI

- Çeşitli veri türleri (tamsayı, gerçek sayı, karakter...) ve veri yapılarını (dizi, kayıt, stack ve kuyruk..) destekleme yeteneği olmalıdır.
- C ve PASCAL veri yapısı bakımından oldukça zengin dillerdir.

GİRİŞ/ÇIKIŞ KOLAYLIĞI

- Sıralı, indexli ve rastgele dosyalara erişme, veritabanı kayıtlarını geri alma, güncelleştirme ve sorgulama yeteneği olarak tanımlanabilir.
- C dili bu bakımdan zayıftır.
- Genel olarak veritabanı programları bu bakımdan güçlüdür.

TAŞINABİLİRLİK

- Taşınabilirlik terimi kaynak kod için kullanılır.
- Bir programlama dilinde yazılmış kaynak kodun başka sistemlerde de sorunsuz derlenerek çalışabilmesine taşınabilirlik denir.
- Seviye düştükçe taşınabilirlik azalır. Taşınabilirliği en az olan diller makine dilleridir.
- C dili taşınabilirliği en yüksek dildir.
- BASIC derleyicileri arasında büyük farklılıklar olduğundan taşınabilirliği yüksek değildir.
- Hiçbir dil için mükemmel taşınabilirlik mümkün değildir

ALTPROGRAMLAMA YETENEĞİ

- Kaynak programların altprogramlara ayrılarak parçalanabilme özelliğidir.

Altprogram kullanmanın faydaları:

- Altprogram kodu küçültür
- Algılamayı daha kolay hale getirir
- Test imkanlarını artırır
- Kaynak kodun güncelleştirilebilirliğini
- Ve yeniden kullanılabilirliğini (reusability)
- Birden fazla kişinin program için çalışabilirliğini sağlar

VERİMLİLİK

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

OKUNABİLİRLİK

- Kaynak kodun çabuk ve kuvvetli bir biçimde algılanabilmesi anlamına gelir.
- **Okunabilirlik** güncelleştirmeyi kolay kılar
- **Okunabilirlik** birçok kişinin ortak kodlar üzerinde birlikte çalışabilmesine imkan sağlar
- **OKUNABİLİRLİK** HİÇBİRŞEYE FEDA EDİLMEMELİDİR

ESNEKLİK

- Bir programlama dilinin programcayı kısıtlamamasına **esneklik** denir.
- **Esnek bir dilde** derleme hataları daha azdır.
- **Esnek bir dilde** birçok işlem hata riskine rağmen programcı için serbest bırakılmıştır.(İyi bir programcı bunu kullanabilir, deneyimsiz ise zararı olabilir)
- C **esnek bir dil**dir. Karakter türü ile tamsayı türü karşılıklı olarak birbirine atanabilir.

ÖĞRENME KOLAYLIĞI

- Her dilin öğrenme zorluğu farklıdır.
- Yüksek seviyeli dillerin öğrenimi daha kolaydır.
- BASIC kolay öğrenilebilen
- C ise öğrenimi kolay olmayan bir dildir
- C++, C# öğrenimi kolay değildir ve önkoşullar gerektirebilir

GENELLİK

- Bir dilin çok çeşitli uygulamalarda etkin olarak kullanılabilmesine **genellik** denir.
- C, PASCAL, BASIC genel dilleri iken
- COBOL, FOXPRO, CLIPPER genelliği olmayan dillerdir.

YAPISALLIK

- Yapısal programlamanın etkin olarak kullanıldığı diller yapısal dillerdir.
- Burada bloklar halinde yazım ön plandadır.
- Yoğun olarak altprogram kullanılmaktadır.
- Program akışında atlamaların yapılması okumayı ve algılamayı zorlaştırabilir.
- Altprogramlar sayesinde soyutlama söz konusudur.

Nesne Yönelimlilik

- Veri+program=nesne yapısına uygun olarak çalışan ve programların nesnelerle yapıldığı dillerdir.
- C# tamamen nesne yönelimli bir dildir.
- C dilinin nesne yönelimli programlama tekniğini destekleyen uyarlaması C++ dilidir.

Dil Seçimini etkileyen Etkenler

Karakteristik	Okunabilirlik	Yazılabilirlik	Güvenilirlik
Sadelik(Simplicity)	X	X	X
Kontrol Yapısı	X	X	X
Veri tip ve yapısı	X	X	X
Syntax tasarım	X	X	X
Soyutlama desteği		X	X
İfade gücü		X	X
Type Checking			X
Exception handling			X
Restricted aliasing			X

Programlama Paradigmaları

- Bir **paradigma**, bir grubun konuya bakış biçimini ve metodunu tanımlayan kavramsal şemadır.
- Programlama paradigmaları, bir programcının problemlere çözüm üretmesini önemli derecede etkilerler.
- Farklı paradigmalar, farklı programlama stilleri getirir ve programcılarının algoritmalara bakış şeklini değiştirirler.

Programlama Paradigmaları

- **Emir Esaslı programlama paradigması**
- **Bildirim Esaslı programlama paradigması**
- **Nesne yönelimli paradigma**
- **Mantık esaslı paradigma**

Imperative Paradigmayı Destekleyen Diller

- *Emir Esaslı (Imperative) paradigma'daki* programlama dilleri işlem tabanlı olup, bir program, bir dizi işlem olarak görülür.
- Programlardaki deyimler, birbirleri ile değişkenler aracılığı ile iletişim kurar.

- Imperative diller, yaygın olarak kullanılan ilk dil grubudur ve günümüzde de yoğun olarak kullanılmaktadır.
- Imperative programlama paradigması, C, FORTRAN, PL/I, Pascal, COBOL, Ada gibi birçok dil tarafından desteklenmektedir.

FORTTRAN ve Pascal, imperative paradigmayı desteklerler.



The diagram illustrates the execution of a sequence of statements in an imperative programming paradigm. It consists of five horizontal, rounded rectangular boxes stacked vertically. Each box has a light orange gradient fill, a dark orange border, and a subtle drop shadow. The boxes are labeled from top to bottom: 'Deyim-1', 'Deyim-2', 'Deyim-3', '...', and 'Deyim-n'. The text is in a bold, blue, sans-serif font. The boxes are set against a light blue background with faint horizontal lines.

Deyim-1

Deyim-2

Deyim-3

...

Deyim-n

Bildirim Esaslı Diller

- Burada bilgisayara bir işlemi nasıl yapacağı bildirilir.
- Veriler ve sonucu elde etmek için *veriye uygulanacak fonksiyonel dönüşümler*, paradigmanın temelini oluşturur.
- Excel bildirim esaslı bir dil olarak düşünülebilir.

Nesneye Yönelik Paradigmayı Destekleyen Diller

- Nesneye yönelik programlama paradigmasının temeli SIMULA 67 programlama dilindedir. Nesnelerin sınıf ve alt sınıflara gruplanması, nesneye yönelik programlamanın temel noktasıdır.
- Smalltalk ve Eiffel gibi diller nesneye yönelik programlama paradigmasını destekler. SIMULA 67, Smalltalk , C++ ve Java

Mantık Paradigmayı Destekleyen Diller

- Mantık programlama paradigmasında programlama, bir işin nasıl yapılacağıının belirtilmesi yerine, ne yapılması istendiğinin belirtilmesi olarak görülür.
- Mantık programlama paradigmasını destekleyen diller, belirli bir koşulun varlığını kontrol ederek ve koşul sağlanıyorsa, uygun bir işlem gerçekleştirerek çalışırlar.
- Bu modeldeki dillere en tanınmış örnek, Prolog programlama dilidir. Mantık tabanlı bir dilin çalışması imperative bir dilin çalışmasına benzemekle birlikte, deyimler sıralı olarak işlenmez.

- Bu dillerin sözdizimi genel olarak şu şekildedir:

koşul_1 -> hareket_1

koşul_2 -> hareket_2

.....

koşul_n -> hareket_n

Paradigma-Yönelik Diller

- Bu bağlamda, bazı diller paradigma bağımsız olup, birden çok paradigmayı destekleyebilirler.
- Örneğin, C++ hem imperative hem de nesneye yönelik programların geliştirilmesini destekler.

C nasıl bir dildir?

- C orta seviyeli bir dildir
- Sistem programlama dilidir
- Algoritmik bir dildir
- Taşınabilirliği en yüksek bir dildir
- İfade gücü yüksek bir dildir
- Okunabilirliği yüksektir
- Çok esnektir
- Verimli bir dildir
- Atomik bir dildir
- Tasarım özellikleri iyi (Güçlü) bir dildir
- Eğitimi zor bir dildir
- Yapısal bir dildir

Özeti

- Bu bölümde programlama dillerinin sınıflandırılması çeşitli yönleriyle incelenmiştir.
- Programlama dillerinin değerlendirilmesinde kullanılan kriterler açıklanmıştır.
- Emir esaslı, nesneye yönelik, fonksiyonel (bildirim esaslı) ve mantık **programlama paradigmaları** açıklanmıştır.