

BLG339
PROGRAMLAMA DİLLERİ KAVRAMI

Hafta 5
Veri Tipleri (Devam)

Yrd. Doç. Dr. Melike Şah Direkođlu

[Alındığı kaynak:](#)

Addison-Wesley's Programming Language Concepts slaytları ve
Prof. Dr. Tuđrul Yılmaz' ın ders notlarından faydalanarak
hazırlanmıştır.

Konular

- Dizi Tipleri
- Kayıt Tipleri
- Birleşik Tipler
- Küme Tipleri
- İşaretçi ve Referans Tipleri

Dizi (Array) Veri Tipi

- Bir dizi aynı tipli veri elemanlarının bir koleksiyonudur.
- İçindeki her bir elemana, ilk elemana göre olan pozisyonuna göre index değeri ile erişilir.
- Örnek:
C: `int aa[4][3][7];`
`toplam += aa[i][j][k];`

Dizi Tasarım Konuları

- İndexler için hangi tipler uygundur?
- İndex elemanlarının değerlerinde sınır kontrolü yapılacak mı?
- İndex aralağı ne zaman sınırlandırılacak?
- İndexler için maksimum değer nedir?
- Bellekte yer ayırma işlemi ne zaman yapılacak?
- Diziye başlangıç değerleri atanabilir mi?
- Diziden herhangi bir tür parça alma mümkün müdür (dilimleme)?

Dizi İndexleme

- İndexleme, index elemanları ile dizi elemanlarını birbirlerine eşler.
 - dizi adı, indeks değeri → bir eleman
- İndeks söz dizim (syntax)
 - Bütün dillerde genel bir kabul görmüş yapı vardır. Önce dizinin adı sonra parantezler veya köşeli parantezler gelir, bunların arasında da index bulunur.
 - FORTRAN, PL/I, Ada **parantez kullanır**.
 - Diğer dillerin çoğu **köşeli parantez kullanır**.

Dizi Index Tipleri ve Aralık Kontrolü

- **Index Tipleri**
 - FORTRAN, C, Java – sadece “integer”
 - Pascal – Her türlü sıralı tip (integer, boolean, char, enum)
 - Ada - integer veya enum (boolean ve char dahil)
- Indexler **C tabanlı dillerde 0’dan başlar**. Fortran’da 1 dir. Bazı dillerde ise tamamen programcı tarafından belirlenir.
- Aralık kontrolü
 - C, C++, Perl, ve Fortran aralık kontrolünü belirtmez.
 - Java, ML, C# aralık kontrolünü belirtir.

İndex Bağlama ve Dizi Kategorileri

- 1. Statik (Durağan)- index limitlerinin bağlanmaları ve dizinin bellekteki bağlanmaları statiktir.
 - FORTRAN 77, Ada'da bazı diziler.
 - C ve C++ dizilerinde static ifadesi varsa statiktir.
 - Avantaj: Yürütme verimi (bellekten yer alma, geri verme yok)

İndex Bağlama ve Dizi Kategorileri

- 2. Sabit Yığıt Dinamik (fixed stack dynamic) - index aralığı statik olarak stack ta (yığıtta) sınırlandırılır, fakat dizinin bellekteki bağlanması (yer ayırma işlemi) tanımlama zamanında dinamik olarak yapılır.
 - C ve C++ dizilerinde static ifadesi yoksa sabit stack (yığıt)-dinamiktir.
 - Avantaj: Bellek verimi

İndex Bağlama ve Dizi Kategorileri

- 3. Yığıt dinamik (stack dynamic) - index aralığı ve dizinin bellekten yer tahsisi çalışma zamanında yapılır.
 - Ada dizileri stack dinamik olabilir.
 - Avantaj: Esneklik - dizinin boyutu kullanılıncaya kadar bilinmek zorunda değildir.

İndex Bağlama ve Dizi Kategorileri

- 4. Sabit Yığın Dinamik (Fixed Heap dynamic) - index bağlanmaları ve dizinin bellekteki bağlanmaları dinamik, fakat bir kez belirlendikten sonra sabit. Bellek geri verilebilir.
 - FORTRAN 90
 - C, C++: malloc, free ...
 - C++: new, delete.
 - Java'da bütün diziler sabit yığın dinamikdir. C# da sabit yığın dinamiği destekler (ArrayList).

İndex Bağlama ve Dizi Kategorileri

- 5. Yığın dinamik (heap dynamic) - index aralığı ve dizinin bellekten yer ayrılması dinamiktir ve istenilen zamanda değiştirilebilir.
 - Avantaj: Esneklik - diziler program çalışması sırasında büyüyebilir veya küçülebilir.
 - APL, Perl ve JavaScript’de diziler isteğe göre büyüyebilir veya küçülebilir.
 - Perl örneği:
 - @list = (1 , 3, 7, 10);
 - push(@list, 13 , 17); // (1 , 3, 7, 10 13 , 17)
 - @list = (); // belleği boşaltır ve iade eder.

Dizi Başlangıç Değerleri

- Genellikle sırayla yazılmış başlangıç değerleri listeleri şeklinde verilir.
- Bazı diller ilklemeye izin verir: C, C++, Java, C#
- Dizi başlangıç örnekleri :
 - C ve C++ - Kıvrık parantezler içinde, derleyici elemanları sayar ve buna göre yer ayarlar:
 - int sayilar [] = {2, 4, 6, 8};
 - char isim[] = “tugrul”; // karakter string
 - char *isimler[] = {“ali”, “veli”} //String dizi (array)
 - Java
 - String[] isimler = [“ali”, “veli”]; //String nesneleri

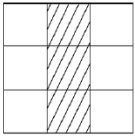
Dizi İşlemleri

- Ada
 - Atama; Sağ tarafta birden çok değer veya diziler olabilir.
 - Birleştirme; bütün tek boyutlu diziler için.
 - İlişkisel operatörler (sadece = ve /=)
- FORTRAN 90
 - Derleyici kütüphanesinden altprogramlarla birçok dizi işlemi; örneğin matris toplamı, çarpımı, vektör iç çarpımı gibi.

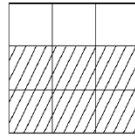
Dizi Dilimleri (Slices)

- Dilimler, dizilerin alt parçalarıdır. Belli bölgelere erişmek için kullanılacak yöntemlerden biridir.
- Dizi işlemleri olan dillerde faydalıdır.
- Dilim örnekleri:
 - 1. FORTRAN 90
 - INTEGER MAT (1:3, 1:3)
 - MAT(1:3, 1) – birinci sütun
 - MAT(2, 1:3) – ikinci satır
 - 2. FORTRAN 95
 - Integer, Dimension (3, 3, 4) :: Cube

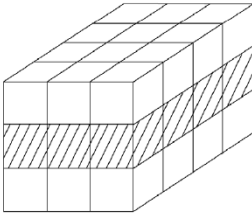
Fortran Örnekleri



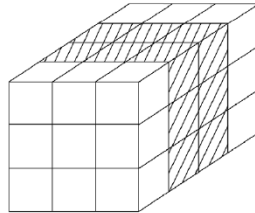
MAT (1:3, 2)



MAT (2:3, 1:3)



CUBE (2, 1:3, 1:4)



CUBE (1:3, 1:3, 2:3)

Dizilerin Gerçekleştirimi

- Dizilere erişimin gerçekleştirilmesi
 - Erişim fonksiyonu indexleri hesaplayarak dizinin içindeki hücrenin bellekte durduğu adrese ulaşır.
 - Bunu yaparken
 - Sütün öncelikli (Fortran) veya
 - Satır öncelikli (Diğer diller) yöntemini kullanır.

Derleme Zamanı Tanımlamaları

Dizilim
Eleman tipi
Altsimge tipi
Altsimge alt limit
Altsimge üst limit
adres

Çok boyutlu dizilim	
Eleman tipi	
Altsimge tipi	
Boyutu = n	
Altsimge 1 alt limit	Altsimge 1 üst limit
...	...
Altsimge n alt limit	Altsimge n üst limit
Adres	

* Altsimge, index demektir.

İlişkili Diziler (Associative Arrays)

- Bir ilişkili dizi kendisiyle aynı sayıdaki anahtar denilen değerlerle indekslenmiş rasgele verilerdir.
- Anahtar değerlerinden bir kıyım (hash) algoritması ile değerler bulunur.
- Tasarım sorunları:
 - 1. Dizilim elemanlarına olan referans şekli nedir?
 - 2. Boyut statik mi yoksa dinamik mi?

Perl de ilişkisel Diziler

- Değişken isimleri “%” ile başlar. Değişken boyu değişkendir.
- Değerler parantezler arasına yazılır:
 - %YukseSicakliklar = (“Pazartesi” => 45, “Salı” => 49,...);
- Indexleme parantezler ve anahtar verilerle yapılır:
 - \$YukseSicakliklar{“Çarşamba”} = 42;
- Elemanlar **delete** ile silinebilir:
 - delete \$YukseSicakliklar{“Salı”};
- Diziyi tamamen boşaltmak için:
 - @YukseSicakliklar = ();


Kayıt Tipleri (Records)

- Bir kayıt (record) heterojen yapıdaki verilerin birleştirilmiş halidir. Her elemanın kendi tipi ve adı vardır.
- Tasarım Kavramları:
 - Alanlara erişim sözdizimsel olarak nasıl yapılacak?

COBOL da Kayıtların Tanımlanması

```
01 ISCI-KAYDI.  
  02 ISCI-ADI  
    05 ILK PICTURE IS X(20)  
    05 ORTA PICTURE IS X(10)  
    05 SOY PICTURE IS X(20)  
02 SAAT-UCRET PICTURE IS 99V99
```

Seviye numaraları



Ada da Kayıtların Tanımlanması

```
type IsciAdiTipi is record  
  ilk: String (1..20);  
  orta: String (1..10);  
  soy: String (1..20);  
end record;  
type IsciKaydiTipi is record  
  IsciAdi: IsciAdiTipi;  
  SaatUcret: Float;  
end record;  
IsciKaydı: IsciKaydiTipi;
```

C Dillerinde Kayıtların Tanımlanması

```

struct IsciAdiTipi{
    char ilk[20];
    char orta[10];
    char soy[20]; }

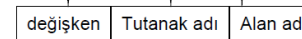
struct IsciKaydiTipi{
    IsciAdiTipi IsciAdi;
    float SaatUcret; }

struct IsciKaydiTipi IsciKaydı;

```

Kayıt Alanlarına Referans

- 1. COBOL seviye numaralarını (level numbers) kullanır
 - field_name OF record_name_1 OF ... OF record_name_n
- 2. Diğerleri diller nokta notasyonu (dot notation) kullanır.
 - record_name_1.record_name_2. ... record_name_n.field_name
- Örnekler:
 - 1. ORTA OF ISCI-ADI OF ISCI-KAYDI (COBOL)
 - 2. IsciKaydi.IsciAdi.orta

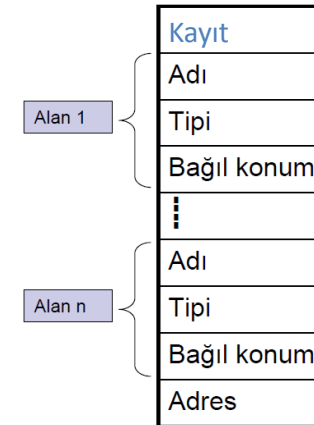


Kayıtlar Üzerindeki İşlemler

- **1. Atama**
 - Pascal, Ada ve C: tipler aynıysa izin verilir.
 - Ada: Sağ taraf sabitler topağı (aggregate constant) olabilir.
- **2. Başlangıç**
 - Ada'da sabitler topağı (aggregate constant) ile.
- **3. Karşılaştırma**
 - Ada'da = ve /=; biri sabitler topağı olabilir.
- **4. MOVE CORRESPONDING**
 - COBOL'da – aynı isimli alanları kopyalar.

Kayıt Tiplerinin Gerçekleştirimi

Her kayıt alanı için, kaydın başlangıcına göre offset adresleri ilişkilendirilir.



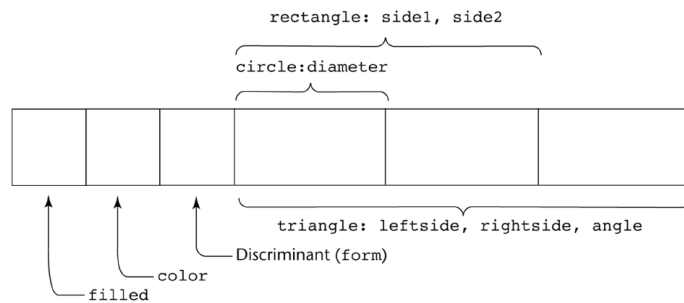
Birleşik Tipler (Unions Types)

- **union tipi:** programın yürütülmesi sırasında değişik zamanlarda değişik tip değerler alabilen değişkenlerdir.
- Tasarım problemleri:
- 1. Bir tip kontrolü yapılacaksa, nasıl bir kontrol yapılacak? Bu kontrolün dinamik olması zorunlu mu?
- 2. Union tipler kayıt tiplerinde kullanılacak mıdır?

Ada Union Tipleri

```
type Shape is (Circle, Triangle, Rectangle);
type Colors is (Red, Green, Blue);
type Figure (Form: Shape) is record
  Filled: Boolean;
  Color: Colors;
  case Form is
    when Circle => Diameter: Float;
    when Triangle =>
      Leftside, Rightside: Integer;
      Angle: Float;
    when Rectangle => Side1, Side2: Integer;
  end case;
end record;
```

Ada Union Tipi



Union ların Değerlendirilmesi

- Potansiyel olarak güvensiz yapılardır.
- Tip kontrolüne izin vermezler.
- Java and C# union ları desteklemez.
 - Artık programlama dillerinde güvenlik önemlidir.