

TÜTEV, D ve C++11 Konferansı

AKAN VERİYİ DİZGİNLEME: Dizi, Dilim ve Şablonlar

Salih Dinçer
salihdb@hotmail.com

Tarih: 30 Haziran 2012

Saat: 12:30

TÜTEV/Ankara

Önce Biraz Tarih...

Sizce bilgisayar (sayma) bilimi ne kadar eskiye dayanıyor olabilir?

- 5 dk.

Not: Milattan önce mi? Örn. 9 bin, 6 bin 500
yoksa sadece 2 bin yıl önce mi?

İşango Kemikleri

(Ishango Bones)

- MÖ. 20000- 18000 arasında kullanılmış olan İşango Kemikleri



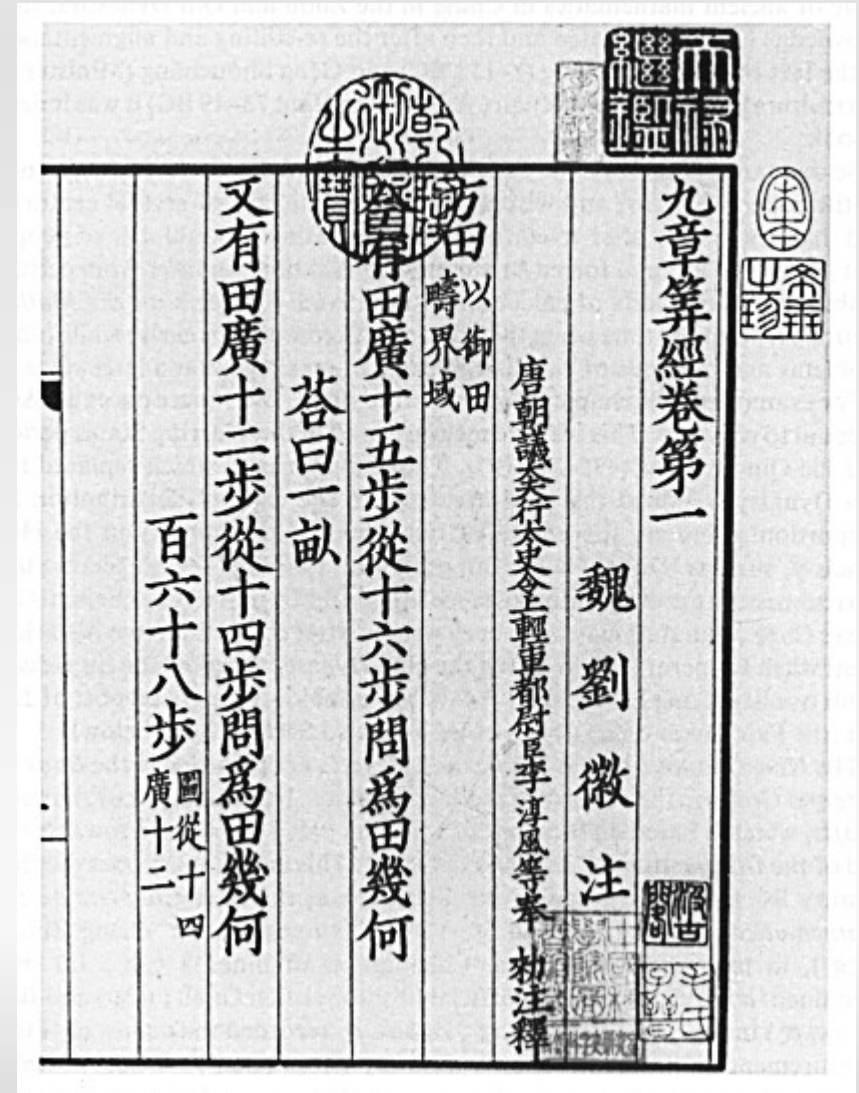
soldakiDizi = [19, 17, 13, 11]; // toplam 60;
ortadakiDizi = [7, 5, 10, 8, 4, 6, 3]; // toplam 48;
sondakiDizi = [9, 19, 21, 11]; // toplam 60;

Matematik Sanatında Dokuz Bölüm (Jiu Zhang Suan Shu)

- MÖ.300 ile MS.200 arasında yazılmış olan Jiu Zhang Suan Shu
- Matrix (Matrices)
- Matris (Dizey)

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

- Köşeli Parantezler
- Çok Boyutlu Diziler



Günümüze dönersek...:)

D Dili'nde diziler ve nitelikleri.

- 20 dk. (x2)

Not: Boyut ~ Uzunluk

D Dizileri

- **İşaretçi (gösterge) Diziler**

`int*[] işaretçiDizi = [&işaretçi1, &işaretçi2, &işaretçi3];`

- **Değişken (tek ve çok boyutlu) Diziler**

- Sabit `int[3] sabitBoyutluDizi = [1, 2, 3];`

- Dinamik `int[] dinamikBoyutluDizi = [1, 2, 3];`

- **Çağrışımsal (associative) Diziler**

(veya anlam olarak "Eşleme Tabloları")

```
int[string] çağrışımsalDizi = [{"bir": 1,  
                                "iki": 2,  
                                "üç": 3  
                                }];
```

Dizilerde İkleme

- D Dilinde, her tür kendi öntanımlı değeriyle ilklenir. Örneğin, int ise **0**, bool ise **false** gibi...
- Öyleyse büyük boyutlu diziler kullanırken bu ilklenme süresini dikkate almalıyız!
- Dizilerde ilklenme değeri nasıl değiştirilir?
 - `bool[3] açıkAnahtarlar = true;`
- Yukarıdaki satır sayesinde tüm dizi elemanları aynı olur. İstersek bunu, derleme anında başka değişkenleri doğrudan eşitleyerek de yapılır:
 - `int[3] öncekiOturum = [cursorX, cursorY, page];`

Dizi Nitelikleri

- `.ptr` ile doğru bir genişlikte adres alma
- `.length`'in çift taraflı kullanımı
- `.sort` ile sıralama
- `.reverse`'de ise tersine sıralama
- `.dup` sayesinde kolay kopyalama
- Yer değiştirme (dilimlere geçtiğimizde!)
- `~` işleci ile birleştirme
- ...

Bölüm Örnekleri

(Veritabanı Sınıfı: ProgramlamaDillerininTarihleri)

- Önce aşağıdaki gibi bir sınıfımız olduğunu farz edelim ve sonra içerdiği bilgiyi basit bir şekilde tüm alt başlıklara sırasıyla uyarlayalım:

```
class ProgramlamaDillerininTarihleri {
    int dilAssembly = 1951;
    int dilB         = 1969;
    int dilC         = 1978;
    int dilD         = 1999;

    string adresliYaz (ref int çıkışYılı) {
        return format(çıkışYılı, " @", &çıkışYılı);
    }
}
```

Bölüm Örnekleri

(Önemli Açıklama)

- Biz bu örnekleri, önceki yansıda yer alan sınıfı, çok karıştırmadan (sanki sınıf içindeymiş gibi) kullanabilmek için aşağıdaki yapı içinde ve ilk satırdaki (import) kütüphaneleri ile birlikte deneyeceğiz...

```
import std.string, std.stdio;
```

```
void main() {  
    with(new ProgramlamaDillerininTarihleri)  
    { // Örnek...  
        : : :  
    }  
}
```

Bölüm Örnekleri

(Sınıfı Tanıma)

```
{ // Örnek 1
    adresliYaz(dilAssembly).writeln;
    adresliYaz(dilB).writeln;
    adresliYaz(dilC).writeln;
    adresliYaz(dilD).writeln;
}
```

```
1951 @B76FCFE8
```

```
1969 @B76FCFEC
```

```
1978 @B76CFF0
```

```
1999 @B76CFF4
```

Bölüm Örnekleri

(Sınıfı Tanıma)

```
import std.string, std.stdio;

class ProgramlamaDillerininTarihleri {
    int dilAssembly = 1951;
    int dilB         = 1969;
    int dilC         = 1978;
    int dilD         = 1999;

    string adresliYaz (ref int çıkışYılı) {
        return format(çıkışYılı, " @", &çıkışYılı);
    }
}

void main() {
    with(new ProgramlamaDillerininTarihleri)
    { // Örnek...
        : : :
    }
}
```

Bölüm Örnekleri

(İşaretçi Diziler ve Diziyi Ters Sıralama)

```
{ // Örnek 2
    int*[] işaretçiDizi = [
        &diAssembly,
        &diB,
        &diC,
        &diD
    ];

    işaretçiDizi.reverse;
    foreach(a; işaretçiDizi) adresliYaz(*a).writeln;
}
```

```
1999 @B76FCFF4
1978 @B76FCFF0
1969 @B76FCFEC
1951 @B76CFE8
```

Bölüm Örnekleri

(Sabit Boyutlu Diziler ve Diziyi Ters Sıralama)

```
{ // Örnek 3
    int[4] sabitBoyutluDizi = [
        dilAssembly,
        dilB,
        dilC,
        dilD
    ];

    sabitBoyutluDizi.writeln("<= Sabit Boyutlu Dizi");
    auto DizininTersi = sabitBoyutluDizi.dup.reverse;
    // DizininTersi.reverse;
} [1951, 1969, 1978, 1999]<= Sabit Boyutlu Dizi
```

Bölüm Örnekleri

(Sabit Boyutlu Diziler ve Diziyi Ters Sıralama)

```
{ // Örnek 4
    int[*][] işaretçiDiziler = [
        &sabitBoyutluDizi[],
        &DizininTersi[]
    ];
    işaretçiDiziler.writeln("<= İşaret Diziler");

    writeln("\t", *işaretçiDiziler[0], "<= 1. elemanı");
    writeln("\t", *işaretçiDiziler[1], "<= 2. elemanı");
}
```

```
[BFA266F8, BFA266E8]<= İşaretçi Diziler
 [1951, 1969, 1978, 1999]<= 1. elemanı
 [1999, 1978, 1969, 1951]<= 2. elemanı
```

Bölüm Örnekleri

(Dinamik Boyutlu Diziler ve Dizi Uzunluğu)

```
{ // Örnek 5
    int[] dinamikDizi = [ dilD ];

    dinamikDizi.write("<= Dinamik Dizi, boyutu: ");
    dinamikDizi.length.writeln;

    dinamikDizi ~= dilC;
    dinamikDizi.write("<= Dinamik Dizi, son boyutu: ");
    dinamikDizi.length.writeln;
}
[1999]<= Dinamik Dizi, boyutu: 1
[1999, 1978]<= Dinamik Dizi, son boyutu: 2
```

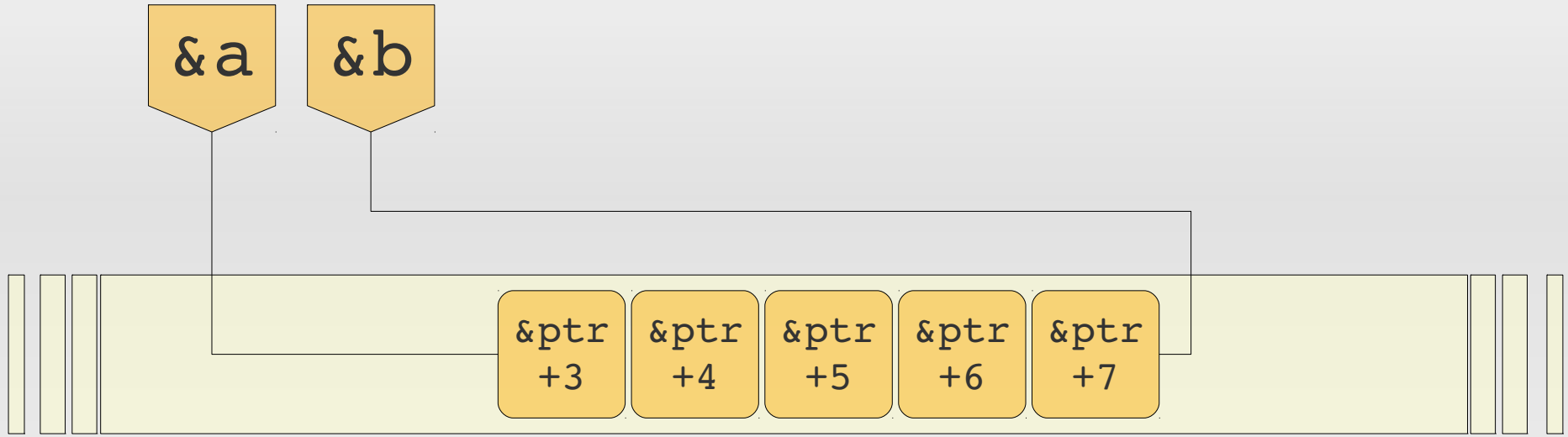

Ara Bölüm

Büyük bir D olanağı: Dilimler

- 15 dk.

Not: $\text{firstIndex} < \text{endIndex}$, Çağrışımsal (associative) dizileri unutulmadı! Son bölümde devam edilecek....:)

Dilimler



- Dilimler, aslında herhangi bir dizinin bir aralığını temsil eden iki işaretçili özel bir değişkendir.
- Erişim sırasında dinamik bir dizi gibi davranırsa da, aslında ayrı ve akıllı bir D olanağıdır.
- Bize başını ve sonunu gösteren bu işaretçiler, dizi elemanlarına dokunmadan erişim sağlar.

Dilimler

```
{ // Örnek 6
    int[] birDilim = sabitBoyutluDizi[0..2];
    birDilim.writeln("<= Bir Dilim [0..2]");

    dinamikDizi ~= birDilim;
    dinamikDizi.writeln("<= Dinamik Dizi");

    int[] başkaBirDilim = sabitBoyutluDizi[2..$];
    başkaBirDilim.writeln("<= Başka bir Dilim [2..$]");

    writeln("\n", birDilim, başkaBirDilim);
}
```

```
[1951, 1969]<= Bir Dilim [0..2]
[1999, 1978, 1951, 1969]<= Dinamik Dizi
[1978, 1999]<= Başka bir Dilim [2..$]

[1951, 1969][1978, 1999]
```

Dilimler

```
{ // Örnek 7a
  int[10] rakamlar = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 ];
  writeln(typeid(rakamlar), &rakamlar);

  int[] ilkYarı = rakamlar [0..$/2];
  ilkYarı.write;
  writeln(typeid(ilkYarı), &ilkYarı);

  int[] ikinciYarı = rakamlar[$/2..$];
  ikinciYarı.write;

  writeln(typeid(ikinciYarı), &ikinciYarı);
}
```

```
int[10]@BFE86630
[1, 2, 3, 4, 5]int[]@BFE86658
[6, 7, 8, 9, 0]int[]@BFE86660
```

Dilimler

```
{ // Örnek 7b
    rakamlar.writeln("(önce)");

    ilkYarı.reverse.write;
    writeln(typeid(ilkYarı), &ilkYarı);

    ikinciYarı.reverse.write;
    writeln(typeid(ikinciYarı), &ikinciYarı);

    rakamlar.writeln("(sonra)");
```

```
} [1, 2, 3, 4, 5, 6, 7, 8, 9, 0] (önce)
   [5, 4, 3, 2, 1]int[]@BFE86658
   [0, 9, 8, 7, 6]int[]@BFE86660
   [5, 4, 3, 2, 1, 0, 9, 8, 7, 6] (sonra)
```

Bölüm Sonu

Çağrışımsal (associative) diziler

- 5 dk.

Not: Hash Table

Çağrışımsal Diziler

(Eşleme Tablosu)

```
string[] günİsimleri = [ "Pazartesi",  
                          "Salı",  
                          "Çarşamba",  
                          "Perşembe",  
                          "Cuma",  
                          "Cumartesi",  
                          "Pazar"  
];
```

```
eleman_türü[indeks_türü] tablo_ismi;
```

Çağrışimsal Diziler

(Eşleme Tablosu)

```
int[string] pazarIlkGün_mü = [ "Pazartesi": 1,  
                                "Salı": 2,  
                                "Çarşamba": 3,  
                                "Perşembe": 4,  
                                "Cuma": 5,  
                                "Cumartesi": 6,  
                                "Pazar": 0  
                                ];
```

```
günİsimleri.writeln;  
pazarIlkGün_mü.writeln;
```

```
["Pazartesi", "Salı", "Çarşamba", "Perşembe", "Cuma",  
 "Cumartesi", "Pazar"]
```


Çağrışimsal Diziler

(Eşleme Tablosu)

```
int[string] pazarIlkGün_mü = [ "Pazartesi": 1,  
                                "Salı": 2,  
                                "Çarşamba": 3,  
                                "Perşembe": 4,  
                                "Cuma": 5,  
                                "Cumartesi": 6,  
                                "Pazar": 0  
                                ];
```

```
günİsimleri.writelN;  
pazarIlkGün_mü.writelN;
```

```
["Cuma":5, "Pazar":0, "Çarşamba":3, "Cumartesi":6,  
 "Salı":2, "Pazartesi":1, "Perşembe":4]
```

Çağrışımsal Diziler

(Tablodan Eleman Çıkarma)

```
int[string] pazarIlkGün_mü = [ "Pazartesi": 1,  
                                "Salı": 2,  
                                "Çarşamba": 3,  
                                "Perşembe": 4,  
                                "Cuma": 5,  
                                "Cumartesi": 6,  
                                "Pazar": 0  
                                ];
```

```
pazarIlkGün_mü.remove("Pazar");  
pazarIlkGün_mü.writeln;
```

```
["Cuma":5, "Pazar":0, "Çarşamba":3, "Cumartesi":6,  
 "Salı":2, "Pazartesi":1, "Perşembe":4]
```

Çağrışımsal Diziler

(Tablodan Eleman Ekleme)

```
int[string] pazarIlkGün_mü = [ "Pazartesi": 1,  
                                "Salı": 2,  
                                "Çarşamba": 3,  
                                "Perşembe": 4,  
                                "Cuma": 5,  
                                "Cumartesi": 6,  
                                "Pazar": 0  
                                ];
```

```
pazarIlkGün_mü ["Pazar"] = 7;  
pazarIlkGün_mü.writeln;
```

```
["Cuma":5, "Pazar":7, "Çarşamba":3, "Cumartesi":6,  
"Salı":2, "Pazartesi":1, "Perşembe":4]
```

Çağrışımsal Diziler

(Diğer Nitelikler)

- **.keys** tabloda bulunan bütün indeksleri dinamik dizi olarak verir.
- **.byKey** tabloda bulunan bütün indeksleri bir aralık olarak sunar.
- **.values** tabloda bulunan bütün elemanları dinamik dizi olarak verir.
- **.byValue** tabloda bulunan bütün elemanları bir aralık olarak sunar.
- **.rehash** ancak gerçekten gereken durumlarda tablonun daha etkin çalışmasını sağlayabilir. Örneğin tabloya çok sayıda eleman eklendikten sonra ve daha tablonun asıl kullanımını başlamadan önce bu nitelik çağrılırsa tablonun erişim işlemleri bazı programlarda daha hızlı olabilir.
- **.sizeof** tablonun referansının büyüklüğüdür (tablodaki eleman adediyle ilgisi yoktur ve her tablo için aynıdır).
- **.get** varsa elemanın değerini, yoksa ikinci parametresinin değerini döndürür.

Teşekkürler...

Sunum daha uzun olacaktı ancak vakit faktörü
bunda etkili oldu!

Herekese dinledikleri için
teşekkür ederim...