

Veri Yapıları ve Algoritmalar

DR. ÖGR. ÜYESİ MEHMET AKİF BÜLBÜL

2023-2024 GÜZ YARIYILI

ALGORİTMİK PROGRAM TASARIMI

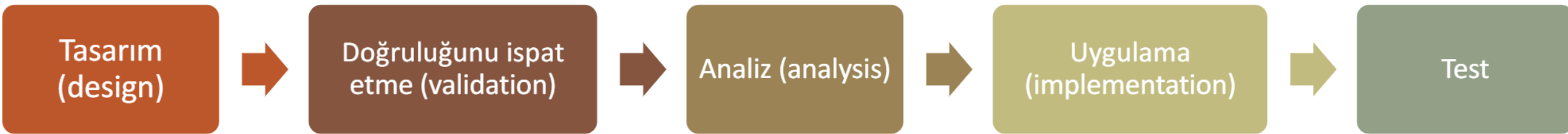
Algoritmik Program Tasarımı Nedir

- Verilen bir problemin bilgisayar ortamında çözülecek biçimde adım adım ortaya koyulması ve herhangi bir programlama aracıyla kodlanması sürecidir.
- Çözüm için yapılması gereken işlemler hiçbir alternatif yoruma izin vermeksizin sözel olarak ifade edilir.
- Verilerin, bilgisayara hangi çevre biriminden girileceğinin, problemin nasıl çözüleceğinin, hangi basamaklardan geçirilerek sonuç alınacağını, sonucun nasıl ve nereye yazılacağını sözel olarak ifade edilmesi biçiminde de tanımlanabilir.

Algoritmanın Önemli Özellikleri

- Algoritma hazırlanırken, çözüm için yapılması gerekli işlemler, öncelik sıraları göz önünde bulundurularak ayrıntılı bir biçimde tanımlanmalıdırlar.
- Yazılan komutun tek bir anlama gelmesi ve herkes tarafından anlaşılır olması gereklidir.
- Yazılan komutların uygulanabilir olması gereklidir.
- Her algoritmanın sonlanması, çalıştırılan komut sayısının sonsuz olmaması gereklidir.

Algoritma Süreci



Kaba-Kod (Pseudo Code)

- Kaba-kod, bir algoritmanın yarı programlama kuralı, yarı konuşma diline dönük olarak ortaya koyulması, tanımlanması, ifade edilmesidir.
- Kaba-kod, çoğunlukla, bir veri yapısına dayandırılmadan algoritmayı genel olarak tasarlamaya yardımcı olur.

Gerçek Kod

- Algoritmanın herhangi bir programlama diliyle, belirli bir veri yapısı üzerinde gerçekleştirilmiş halidir.
- Bir algoritmanın gerçek kodu, yalnızca, tasarlandığı veri yapısı üzerinde çalışır.
- Bir algoritma kaba-kod ile verilirse gerçek kod verilmesinden daha kolay anlaşılır.

Kaba Kod: Temel Gösterim

1. Bir değer atamak için genellikle `:=` veya `<-` kullanılır. `=` işareti ise eşitlik kontrolü için kullanılır.
2. Metot, fonksiyon, yordam isimleri: Algoritma Adı ({parametre listesi})
3. Program yapısı şu şekilde tanımlanır::
 - **Karar yapıları:** `if ... then ... else ...`
 - **while döngüleri:** `while ... do {döngü gövdesi}`
 - **Tekrar döngüleri:** `repeat {döngü gövdesi} until ...`
 - **for döngüleri:** `for ... do {döngü gövdesi}`
 - **Dizi indeksleri:** `A[i]`
4. Metotların çağırılması: Metot adı ({değişken listesi})
5. Metotlardan geri dönüş: `return` değer

Kaba Kod: Temel Gösterim

```
IF mutluyusan THEN
    gülümse
ELSE IF üzgünsen
    kaşlarını çat
ELSE
    normal yüz ifadesi
ENDIF
```

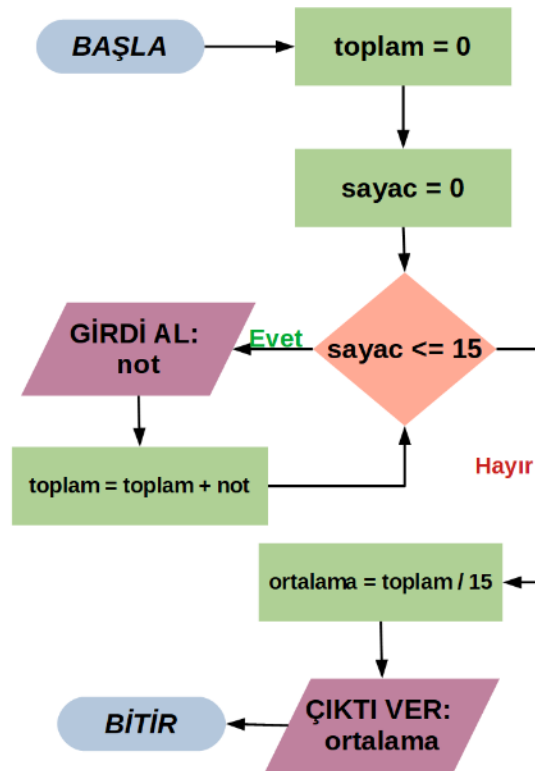
```
INPUT renk
CASE renk
    red: PRINT "kırmızı"
    green: PRINT "yeşil"
    blue: PRINT "mavi"
OTHERS
    PRINT "bilinmeyen renk"
ENDCASE
```


Kaba Kod: Temel Gösterim

```
FOR yıl içindeki her bir ay için  
    Compute gün sayısını hesapla  
ENDFOR
```

```
WHILE Nufus_sayısı < Limit  
    Compute Nufus_sayısı as  $Nufus\_sayısı + Dogum - Ölüm$   
ENDWHILE
```

Kaba Kod: Temel Gösterim



$\text{toplam} \leftarrow 0$

$\text{sayac} \leftarrow 0$

WHILE $\text{sayac} \leq 15$

$\text{not} \leftarrow \text{INPUT}$

$\text{toplam} \leftarrow \text{toplam} + \text{not}$

$\text{ortalama} \leftarrow \text{toplam} / 15$

PRINT ortalama

Kaba Kod: Temel Gösterim

Örnek: Bir dizideki elemanların toplam ve çarpımını hesaplayan algoritmayı kaba-kod kullanarak tanımlayınız?

Kaba Kod: Temel Gösterim

Toplam_Carpim_Hesapla(dizi[n])

Input: n adet tam sayıdan oluşan dizi

Output: dizi elemanlarının toplam ve çarpım sonucu

toplam <- 0

carpim <- 1

FOR i:= 1 TO n DO

 toplam <- toplam + dizi[i]

 carpim <- carpim * dizi[i]

ENDFOR

RETURN (toplam, carpim)

Kaba Kod: Temel Gösterim

Örnek: Bir sayının faktöriyelini hesaplayan algoritmayı kaba-kod kullanarak tanımlayınız?

Kaba Kod: Temel Gösterim

Bir dizinin aritmetik ortalamasını bulan ve sonucu geri gönderen «ortalama_bul» isimli fonksiyonun kaba kodunu ve C dili ile programını yazınız?

Kaba Kod: Temel Gösterim

ortalama_bul(dizi[n])

Input: n adet tam sayıdan oluşan dizi

Output: dizi elemanlarının ortalaması

toplam <- 0

Sonuc<-0

FOR i:= 1 TO n DO

 toplam <- toplam + dizi[i]

ENDFOR

Sonuc=toplam/n

RETURN (Sonuc)

Kaba Kod: Temel Gösterim

```
float ortalama_bul(float A[], int n)
{
    float ortalama, toplam=0;
    int k;
    for(k=0; k<n; k++)
        toplam=toplam+A[k];
    ortalama=toplam/n;
    return ortalama;
}
```