

# PROGRAMLAMA TEMELLERİ

Algoritma Oluřturma

# Algoritmanın Gösterimi

Kodlamadan önce düşünülen algoritma kağıda dökülür.

En yaygın 3 gösterim şekli

- İşlem adımları (process steps)
- Söзде kod / Kodsuz (pseudo code)
- Akış diyagramı (flowchart)

# İşlem adımları

Genellikle tek düze çözümlerin ifadesinde kullanılır.

Her satırda bir işlem adımı gösterilir.

Konuşma dilinde yazılır.

## Yarıçapı verilen dairenin alanını hesaplama

Adım 1- Dairenin yarıçapını oku.

Adım 2- Yarıçapın karesi ile 3,1416'yı çarp.

Adım 3- Sonucu ekrana yaz.

# İşlem adımları

## Klavyeden girilen 10 sayının ortalamasını hesaplama

Adım 1- **Toplamı** ve **Sayacı** sıfırla.

Adım 2- Bir **sayı** oku.

Adım 3- **Sayacı** “1” artır.

Adım 4- Okuduğun **sayıyı** **toplama** ekle.

Adım 5- **Sayac** “10” dan küçükse 2. Adıma git.

Adım 6- **Toplamı** “10” a böl, **ortalama** olarak kaydet.

Adım 7- **Ortalamayı** yaz.

# İşlem adımları

## Avantajları

- 👍 Konuşma dilinde yazılır.
- 👍 Özel bir kodlama öğrenilmesi gerekmez.
- 👍 Okuması ve yazması kolaydır.

## Dezavantajları

- 👎 Belli bir standardı ve sistematığı yoktur.
- 👎 Algoritmanın metinsel açıklamasından farksızdır.
- 👎 Sadece yazıldığı dili anlayan kişilerce okunabilir.
- 👎 Karmaşıklığı ve yuvalanmalar görünmez.



# Sözde Kod

İşlem adımlarının programlama diline benzetilmesi ile oluşturulmuştur.

Konuşma diline yakındır (İngilizce konuşuyorsanız).

Daha sistemattir.

**Yarıçapı verilen dairenin alanını hesaplama**

```
READ yaricap
```

```
alan ← 3,1416 * yaricap ^2
```

```
WRITE alan
```

# Sözde Kod

Klavyeden girilen 10 sayının ortalamasını hesaplama

**Toplam**  $\leftarrow$  0

**Sayac**  $\leftarrow$  0

**REPEAT**

**READ** **sayi**

**Sayac**  $\leftarrow$  **Sayac** +1

**Toplam**  $\leftarrow$  **Toplam** + **sayi**

**UNTIL** **Sayac** = 10

**Ortalama**  $\leftarrow$  **Toplam** / 10

**WRITE** **Ortalama**

# Sözde Kod

## Avantajları

- ✓ Basit sözde kod kurallarını öğrenen herkes anlayabilir.
- ✓ Kodlamaya geçişi kolaydır.
- ✓ PASCAL diline benzer (PASCAL ile kodlayacaksanız.)
- ✓ Girintiler sayesinde döngü ve yuvalanmalar fark edilir.
- ✓ Evrenseldir, her millet anlayabilir.



## Dezavantajları

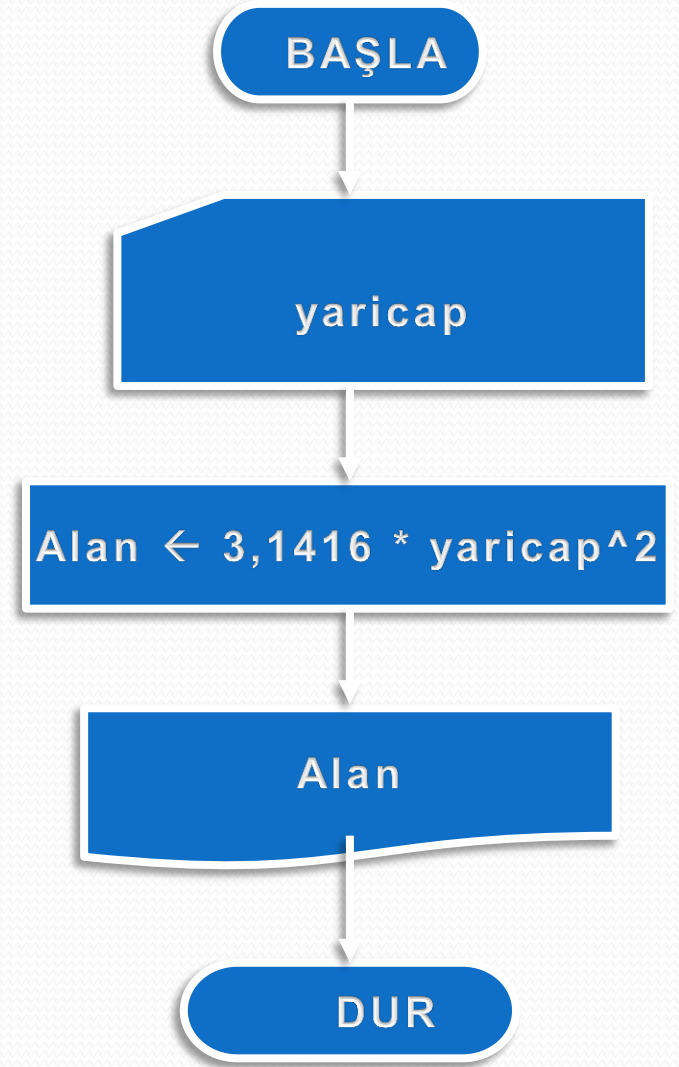
- ✗ Bazı temel komutları bilmek gerekir.
- ✗ Programlama mantığını ve kontrol deyimlerinin çalışma prensibini bilmek gerekir.





# Akış diyagramı

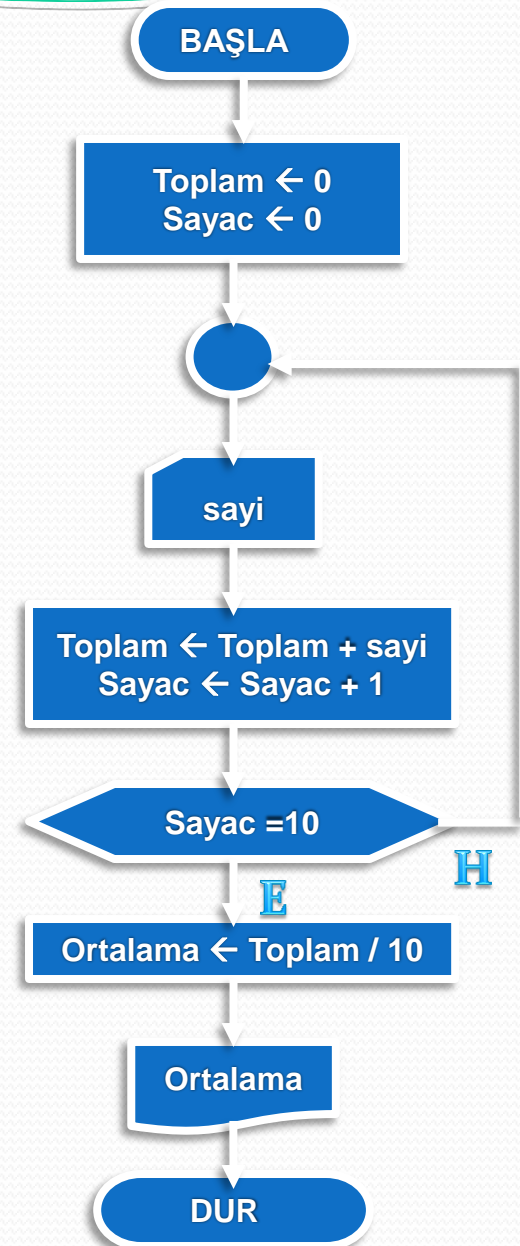
- Algoritmanın görsel sunumudur.
- Anlaması ve anlatması kolaydır.
- Çok sistemattiktir.



Yarıçapı verilen dairenin alanını hesaplama

# Akış diyagramı

Klavyeden girilen 10 sayının ortalamasını hesaplama



# Akış Diyagramı

## Avantajları

- 😊 Görseldir.
- 😊 Yuvalanmalar, döngüler, akış kontrolleri kolayca takip edilebilir.
- 😊 Dilden bağımsızdır, her millet anlayabilir. Şekillerde ufak tefek farklılıklar olsa da fonksiyonu anlaşılır.
- 😊 Kodlamaya geçişi kolaydır.

## Dezavantajları

- 😞 Şekillerin anlamları öğrenilmelidir.
- 😞 (Her ne kadar flowchart çizen programlar olsa da) Çizim için zaman ayırmak gerekir.

