



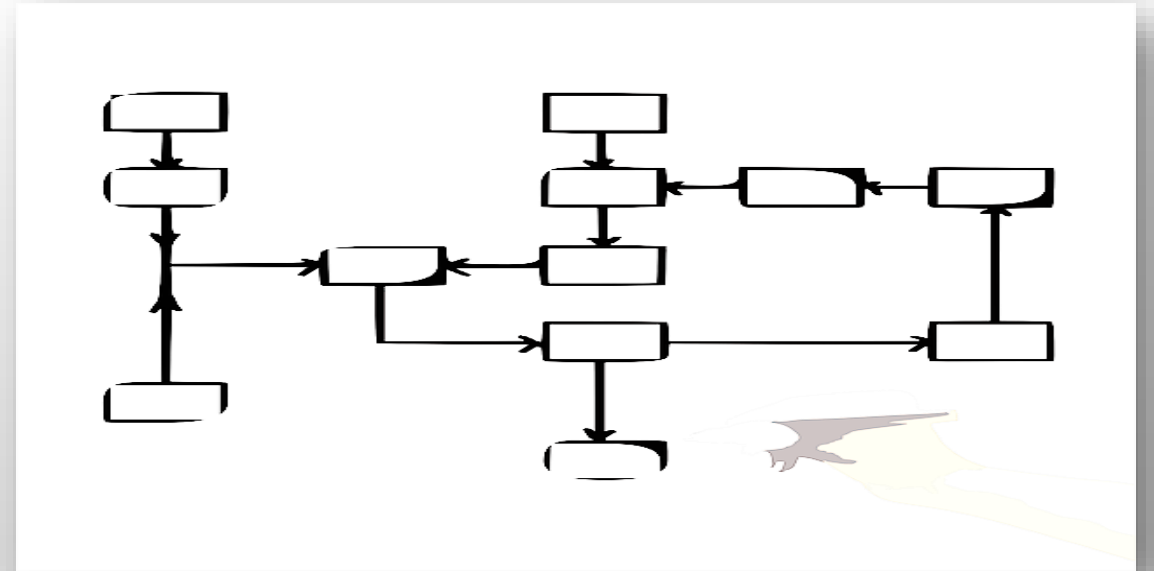
# **Bölüm 1: Akış Diyagramları**

## **JAVA ile Nesne Yönelimli Programlama**



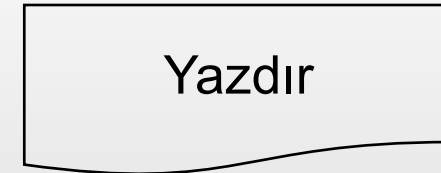
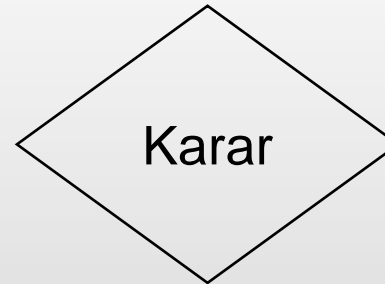
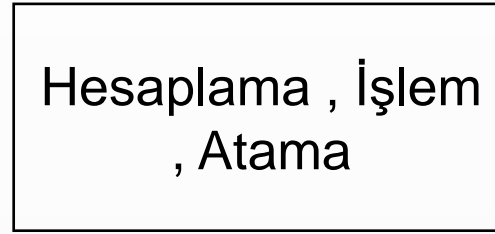
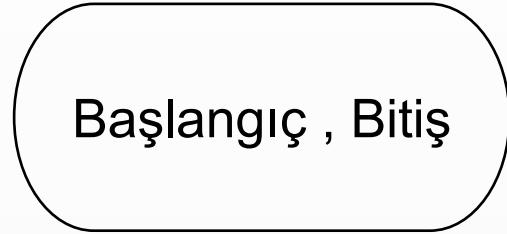
# Akış Diyagramları (Flowcharts)

- **Algoritma**, bir problemi çözmek veya belirli bir görevi gerçekleştirmek için adım adım talimatlar içeren bir plan.
- **Akış Diyagramları**, algoritmanın adımlarını, adımlar arasındaki bağlantıları ve kontrol mekanizmalarını görsel bir şekilde temsil eder.





# Temel Elemanlar





# Nasıl Oluşturulur?

- İşlemler adım adım belirlenir.
- Adımlar mantıklı bir sırayla yerleştirilir.
- Tüm işlemler akış çizgileri ile birbirine bağlanır.
- Koşullu ifadeler ile karar noktaları belirlenir.



# Faydaları

- Karmaşık süreçleri basitleştirir.
- Hata ve iyileştirmeleri belirlemeye yardımcı olur.
- Süreçleri belgelemede, raporlamada kullanılır.



# Kontrol Mekanizmaları

- Algoritma işlem adımları sıralı bir şekilde gerçekleşir.
- Koşullara bağlı olarak farklı işlem adımlarına yönelinebilir.
- Belirli bir koşul sağlandığı sürece bazı adımlar tekrarlanır.



# Sıralı Akış (Sequential Flow)

- Yapılacak işlerin (komutlar) art arda mantıksal bir sıra ile dizilmesidir.
- **Basit:** Sade ve anlaşılır bir akışı temsil eder.
- **Kolay Bakım:** Hata ayıklama ve kodun anlaşılması kolaydır.
- **Bağlamdan Bağımsız:** İşlemler bağımsız olarak sıralanabilir.
- **Geri Dönüş Yok:** Bir işlem başladığında geri dönüş mümkün değildir.

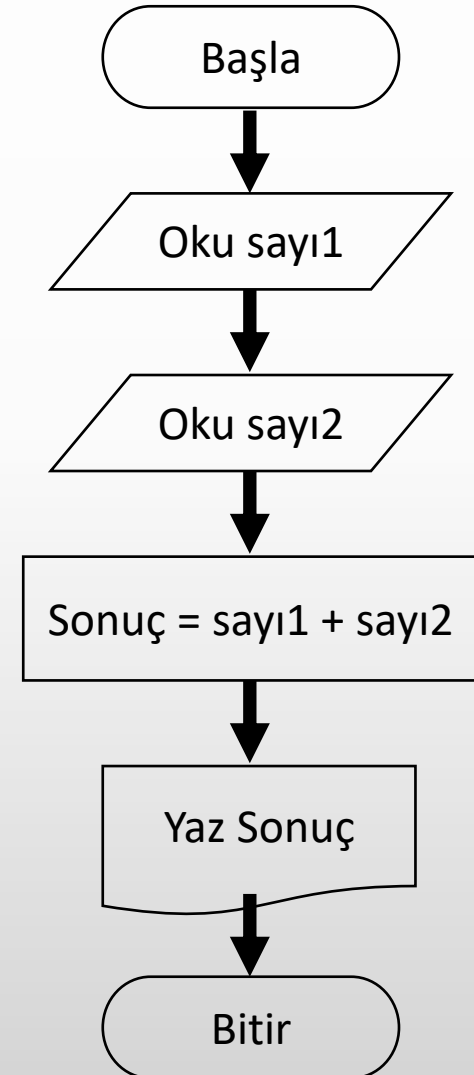


# Sıralı Akış (Sequential Flow)

- Klavyeden girilen iki sayının toplamını ekrana yazma

- **Algoritma**

1. Başla
2. İlk sayıyı oku
3. İkinci sayıyı oku
4. İki sayıyı topla, sonuç'a ata
5. Sonuç'u ekrana yaz
6. Bitir







# Koşullu Akış (Conditional Flow)

- İhtimallerden birinin seçildiği akış türüdür.
- **Karar Noktası (K):** Bir koşul sorusu bulunur.
- **Evet'e Göre Akış (P):** Sorunun cevabı evet ise, P işlemi yönünde bir akış
- **Hayır'a Göre Akış (Q):** Cevap hayırsa, Q işlemi yönünde bir akış izlenir.
- **Esneklik:** Koşullara göre farklı şekillerde davranabilme yeteneği.
- **Karar Verme Yetisi:** Koşullara bağlı işlemleri gerçekleştirme yetisi.
- **Kod Kontrolü:** Durumları kontrol etmek ve buna göre işlem yapmak.

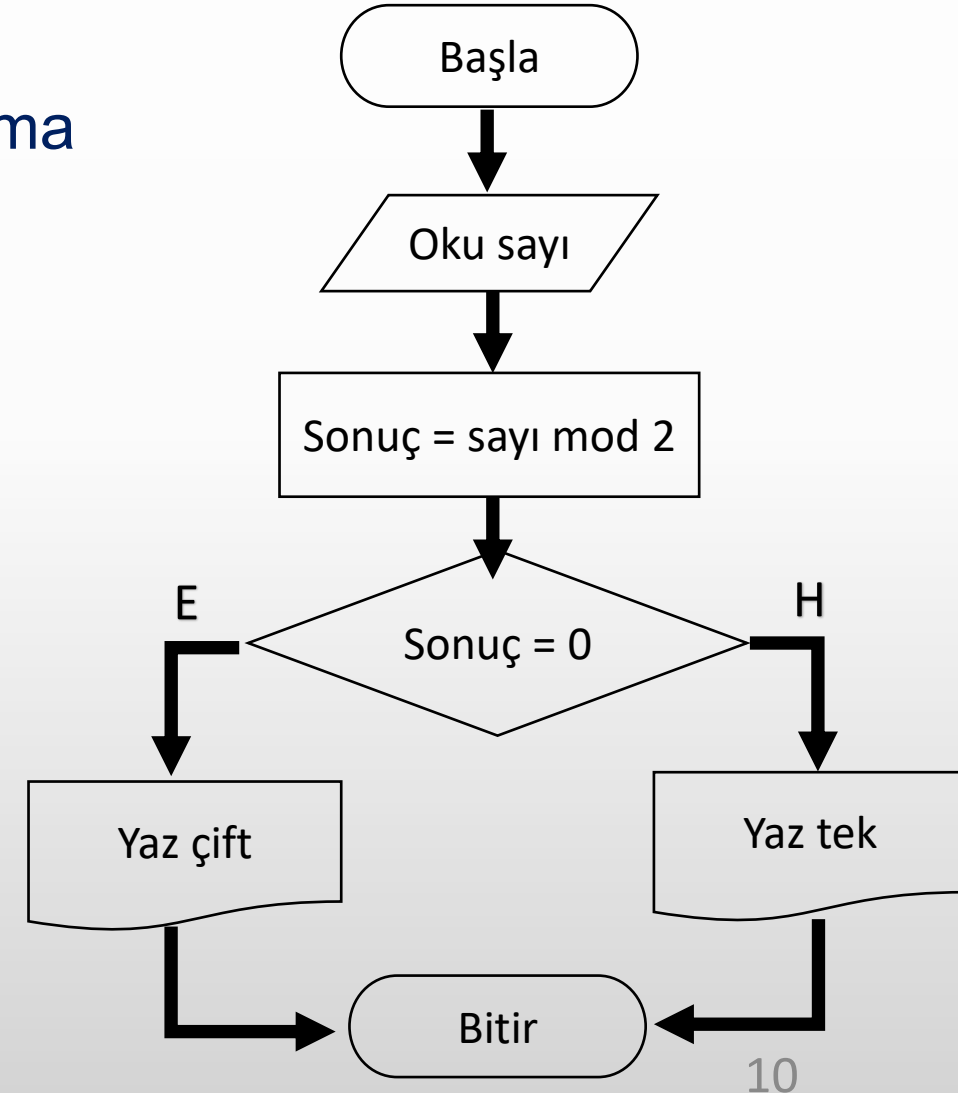


# Koşullu Akış (Conditional Flow)

- Verilen sayının tek/çift olduğunu ekrana yazdırma

- Algoritma**

1. Başla
2. Sayıyı oku
3. Sayı mod 2 al, sonuç'a ata
4. Sonuç 0 ise 5'e git, değilse 6'ya git
5. Ekrana çift yaz, 7'ye git
6. Ekrana tek yaz
7. Bitir





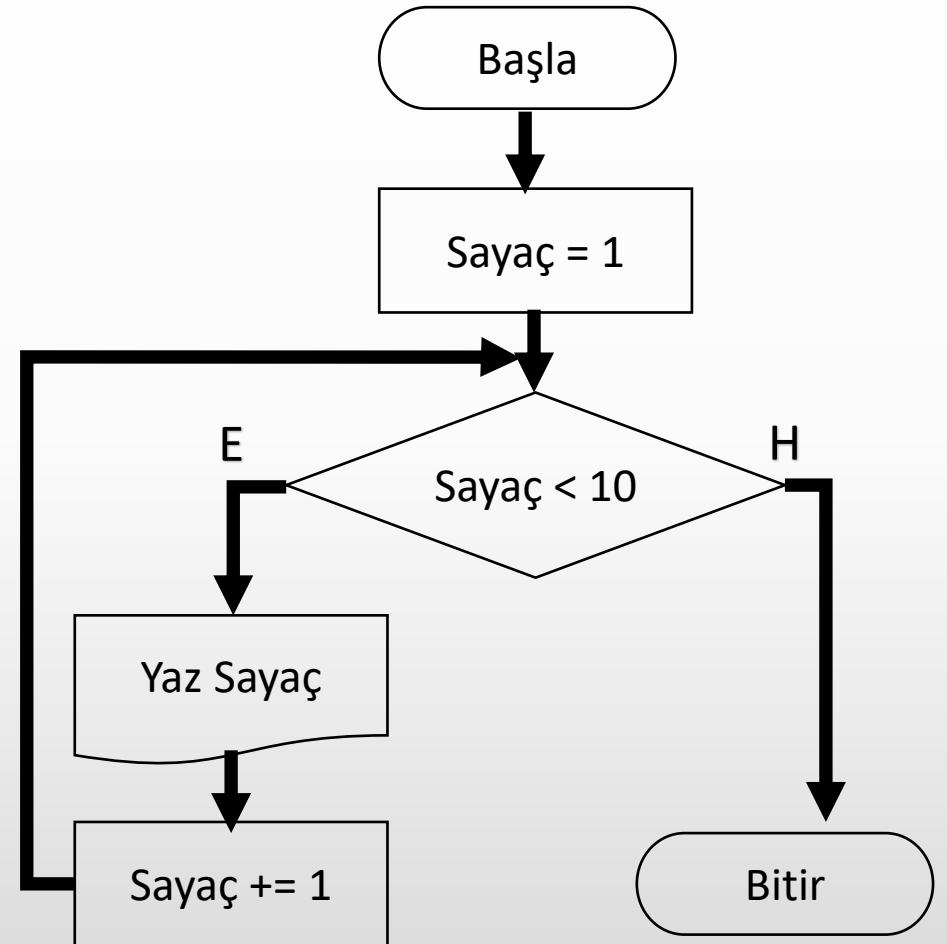
# Tekrarlı Akış

- İşlemlerin birden çok defa tekrar ettiği akış şemalarıdır.
- **Koşullara Bağlı Döngüler:** Koşul sağlandığı sürece tekrar eder.
- **Belirli Sayıda Tekrarlar:** Belirli bir sayıda tekrarın gerçekleşmesini sağlar.
- **Sonsuz Döngüler:** Koşul sürekli sağlandığı için işlemler sürekli tekrarlar.
- **Verimli:** Belirli işlemleri tekrar etme ve kodu optimize etme yeteneği.
- **Dinamik:** Programın değişen koşullara uyum sağlama yeteneği.
- **Kod Tekrarını Önleme:** İşlemleri tekrar etmek için kodun tekrar yazılmasına gerek yoktur.



# Tekrarlı Akış

- 1'den 10'a kadar olan sayıları yazdırma
- **Algoritma**
  1. Başla
  2. Sayaç = 1
  3. 10'dan küçükse, 4'e git, değilse 7'ye git
  4. Yaz sayaç
  5. Sayaç 1 arttır
  6. 3'e git
  7. Bitir



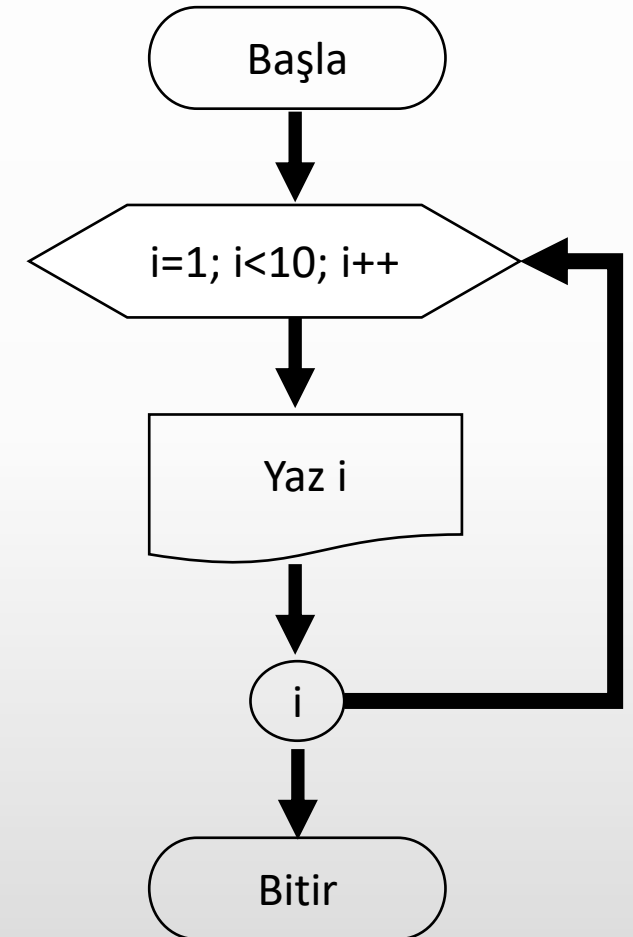


# Döngüye Özel Akış

- 1'den 10'a kadar olan sayıları yazdırma

- **Algoritma**

1. Başla
2. Döngü 1'den 10'a kadar
3. Yaz döngü değişkeni
4. Döngü sonu
5. Bitir





SON