



# **Bölüm 1: Akış Diyagramları**

## **JAVA ile Nesne Yönelimli Programlama**

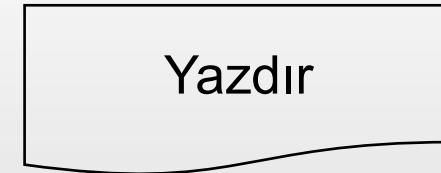
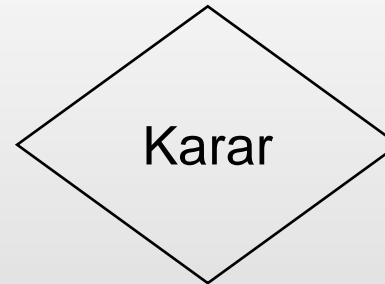
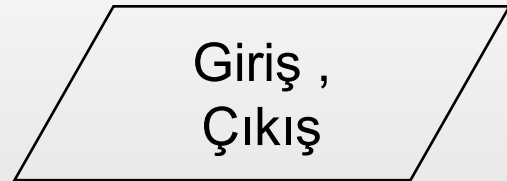
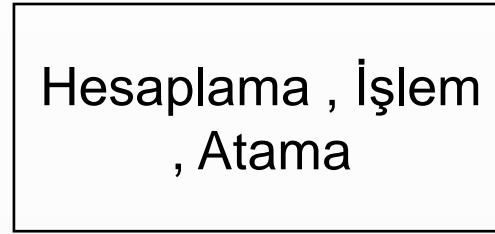
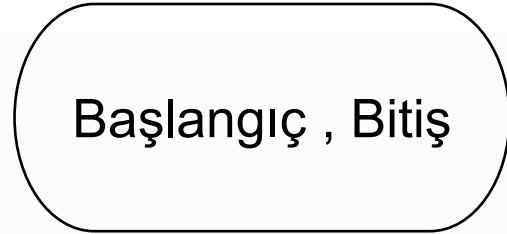


# Akış Diyagramları (Flowcharts)

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



# Temel Elemanlar





# Nasıl Oluşturulur?

- **Adımları Belirle:** İşlemleri adım adım belirle.
- **Sıralamayı Oluştur:** Adımları mantıklı bir sırayla yerleştir.
- **Bağlantıları Ekle:** İşlemleri akış çizgileri ile bağla.
- **Karar Noktalarını Ekle:** Koşullu ifadeleri belirle.



# Faydaları

- **Anlaşılabilirlik:** Kompleks süreçleri basitleştirir.
- **Analiz:** Hataları ve iyileştirme fırsatlarını belirlemeye yardımcı olur.
- **Dokümantasyon:** Süreçleri belgelemede kullanılır.



# Kontrol Mekanizmaları

- **Sıralı İşlem:** Adımlar sıralı bir şekilde gerçekleşir.
- **Karar Yapıları:** Koşullara bağlı olarak farklı adımlara yönlendirme yapar.
- **Döngüler:** Belirli bir koşul sağlandığı sürece adımları tekrarlar.



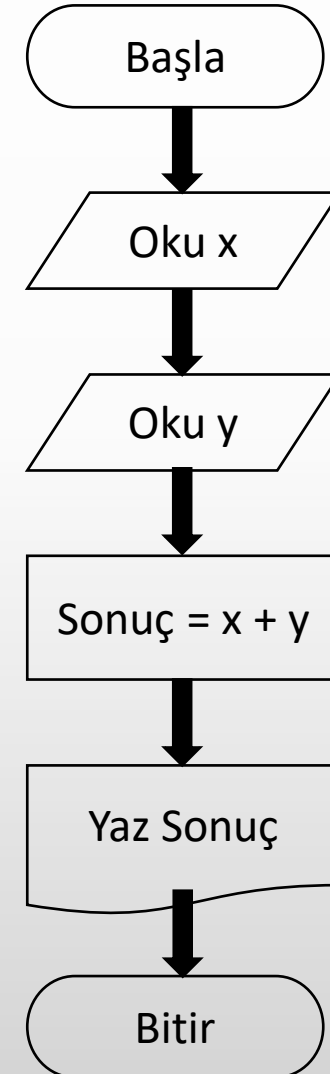
# Sıralı Akış (Sequential Flow)

- Yapılacak işlerin (komutların) art arda mantıksal bir sıra ile dizilmesidir.
- **Avantajları**
  - **Basitlik:** Sade ve anlaşılır bir akışı temsil eder.
  - **Kolay Bakım:** Hata ayıklama ve kodun anlaşılması daha kolaydır.
  - **Bağlam Bağımsızlık:** İşlemler birbirinden bağımsız olarak sıralanabilir.
- **Dikkat Edilmesi Gerekenler**
  - **Geri Dönüş Yok:** Bir işlem başladığında geri dönüş mümkün değildir.
  - **Kararlılık:** İşlemlerin sırasının doğru ve istenen sonuca götüreceğinden emin olun.



# Sıralı Akış (Sequential Flow)

- Klavyeden girilen iki sayının toplamını ekrana yazma
- **Algoritma**
  - Başla
  - İlk sayıyı oku
  - İkinci sayıyı oku
  - İki sayıyı topla, sonuç'a ata
  - Sonuç'u ekrana yaz
  - Bitir







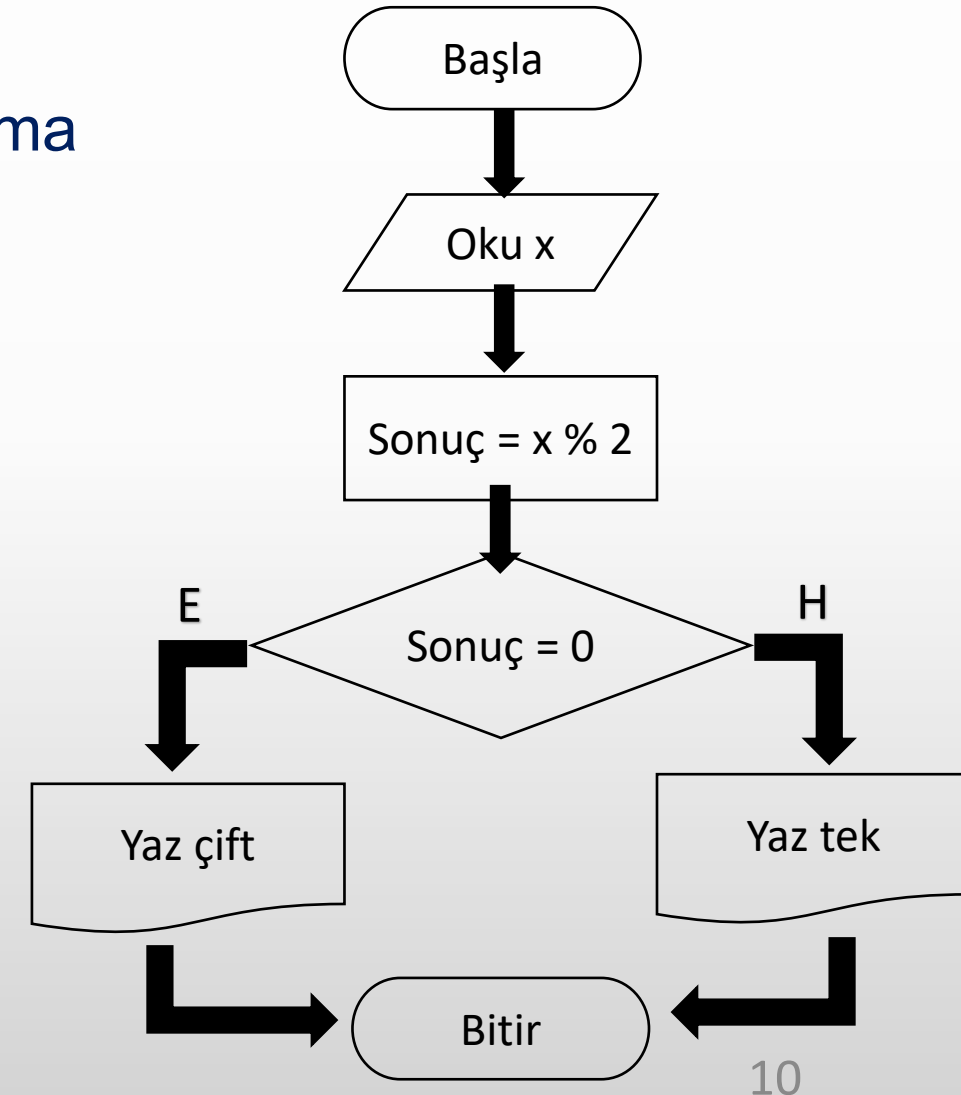
# Şartlı Akış (Conditional Flow)

- Programda seçeneklerden birini seçme durumunun yer aldığı akış türüdür.
- **Karar Noktası (K):** Bir koşul sorusu bulunur.
- **Evete Göre Akış (P):** Sorunun cevabı evet ise, P işlemi yönünde bir akış
- **Hayıra Göre Akış (Q):** Cevap hayırsa, Q işlemi yönünde bir akış izlenir.
- **Avantajları**
  - **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.



# Şartlı Akış (Conditional Flow)

- Verilen sayının tek/çift olduğunu ekrana yazdırma
- **Algoritma**
  - Başla
  - Sayıyı oku
  - Sayı mod 2 al, sonuç'a ata
  - Sonuç 0 ise 5'e git, değilse 6'ya git
  - Ekrana çift yaz, 7'ye git
  - Ekrana tek yaz
  - Bitir





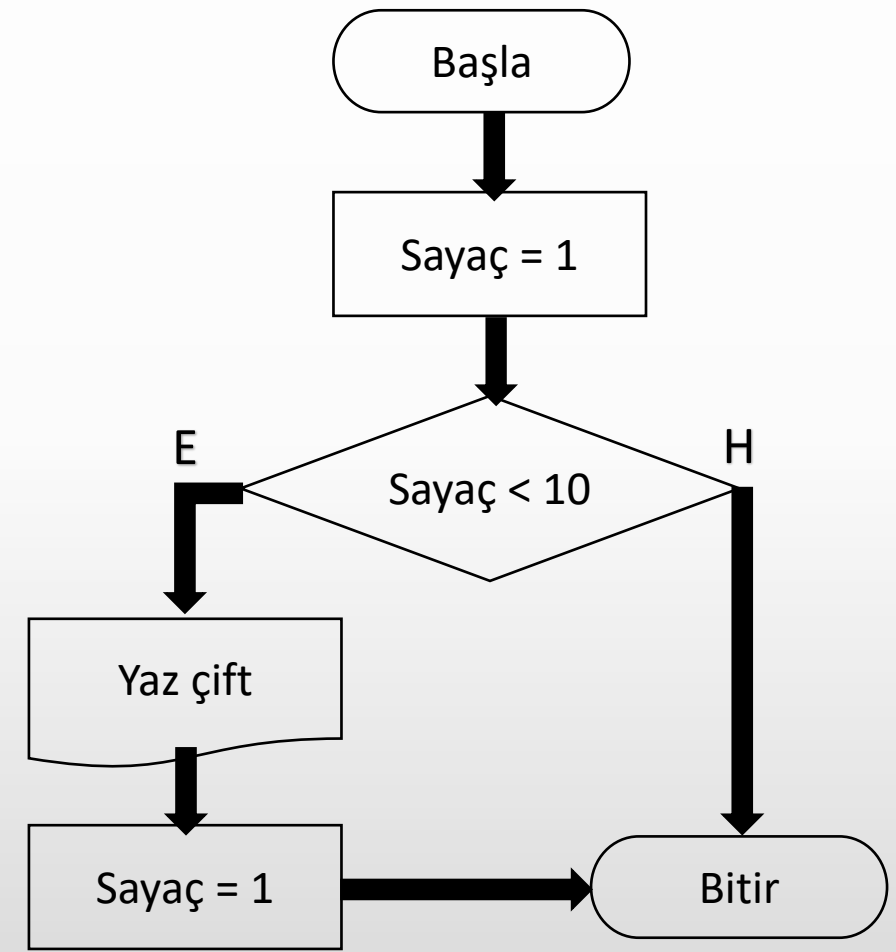
# Tekrarlı Akış

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



# Tekrarlı Akış

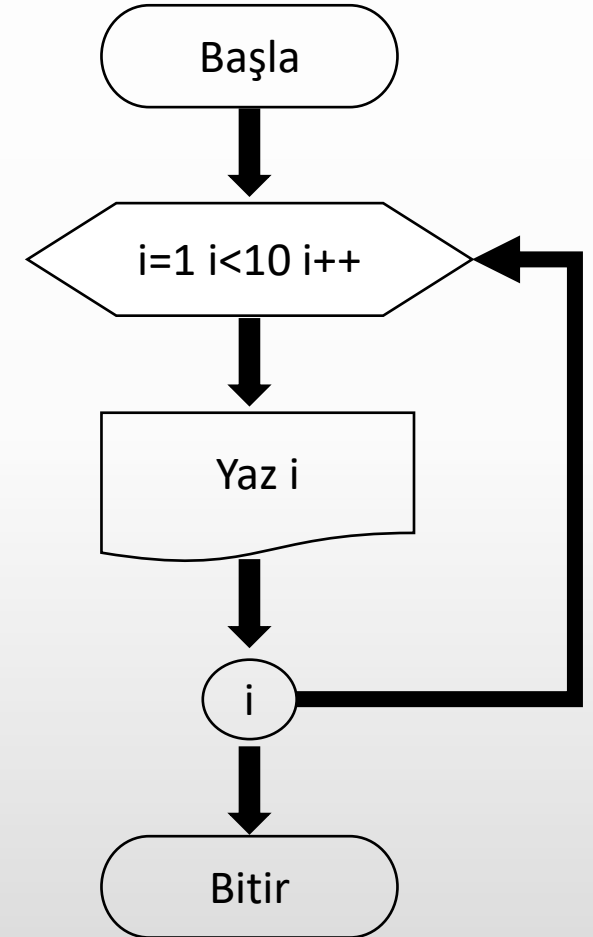
- 1'den 10'a kadar olan sayıları yazdırma
- **Algoritma**
  - Başla
  - Sayaç = 1
  - 10'dan küçükse, 4'e git, değilse 7'ye git
  - Yaz sayaç
  - Sayaç 1 arttır
  - 3'e git
  - Bitir





# Döngüye Özel Akış

- 1'den 10'a kadar olan sayıları yazdırma
- **Algoritma**
  - Başla
  - Döngü 1'den 10'a kadar
  - Yaz sayaç
  - Döngü sonu
  - Bitir





SON