



Bölüm 1: Akış Diyagramları

JAVA ile Nesne Yönelimli Programlama



Temel Elemanlar

Başlangıç , Bitiş

Hesaplama , İşlem
, Atama

Döngü

Giriş ,
Çıkış

Karar

Yazdır



Nasıl Oluřturulur?

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



Faydaları

- Karmaşık süreçleri basitleştirir.
- Hataları ve iyileştirme fırsatlarını belirlemeye yardımcı olur.
- Süreçleri belgelemede, raporlamada kullanılır.



Kontrol Mekanizmaları

- Algoritma adımları sıralı bir şekilde gerçekleşir.
- Koşullara bağlı olarak farklı adımlara yönlendirilir.
- Belirli bir koşul sağlandığı sürece bazı adımlar tekrarlanır.



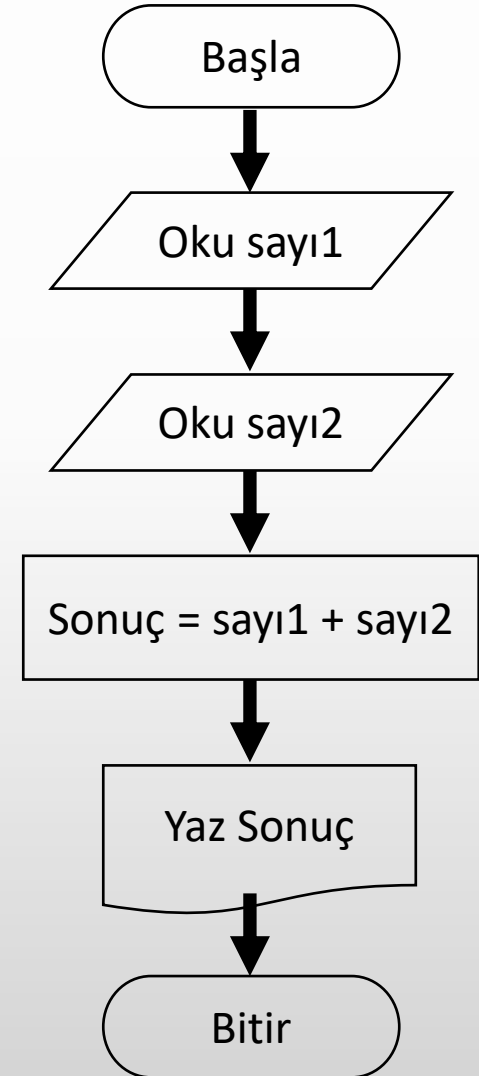
Sıralı Akış (Sequential Flow)

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





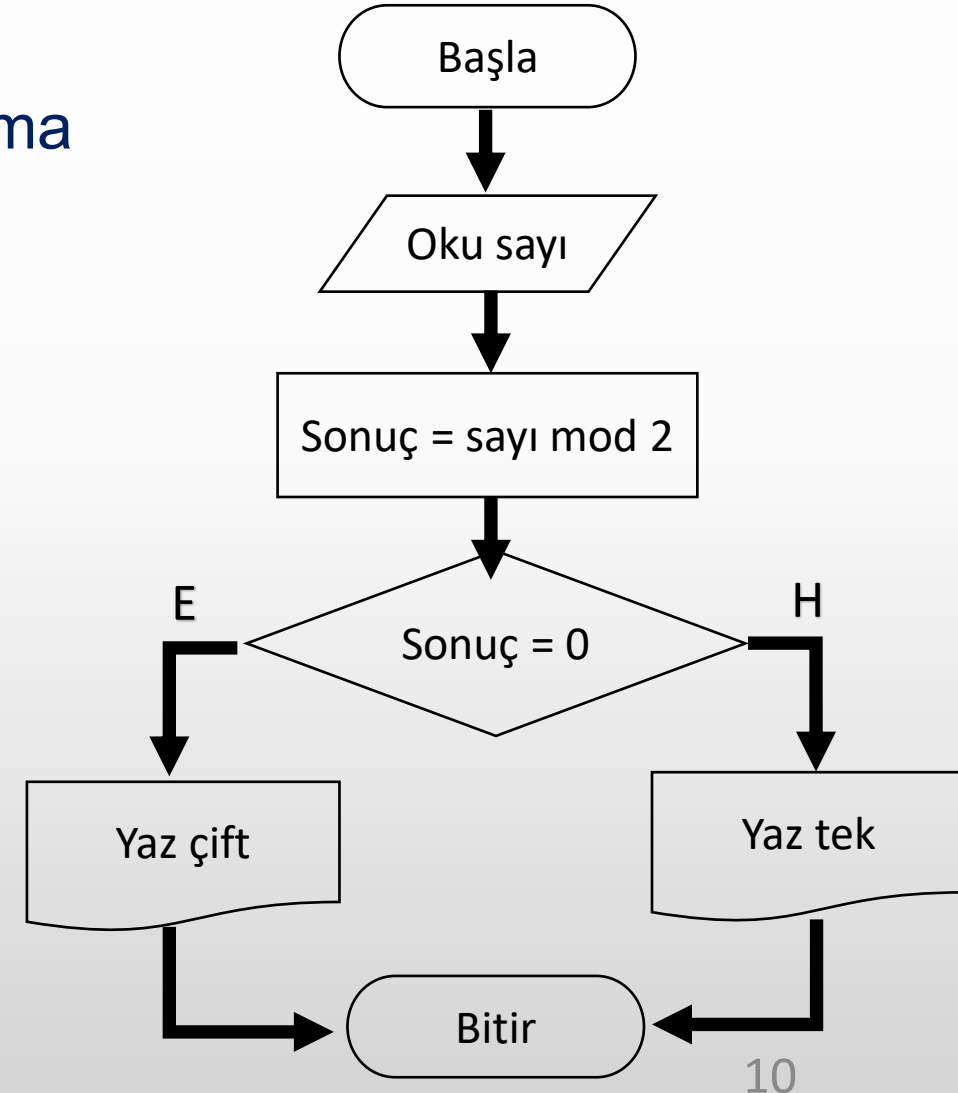
Şartlı Akış (Conditional Flow)

- İhtimallerden birini seçme durumunun yer aldığı 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.
- **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**
 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.
- **Belirli 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.
- **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.

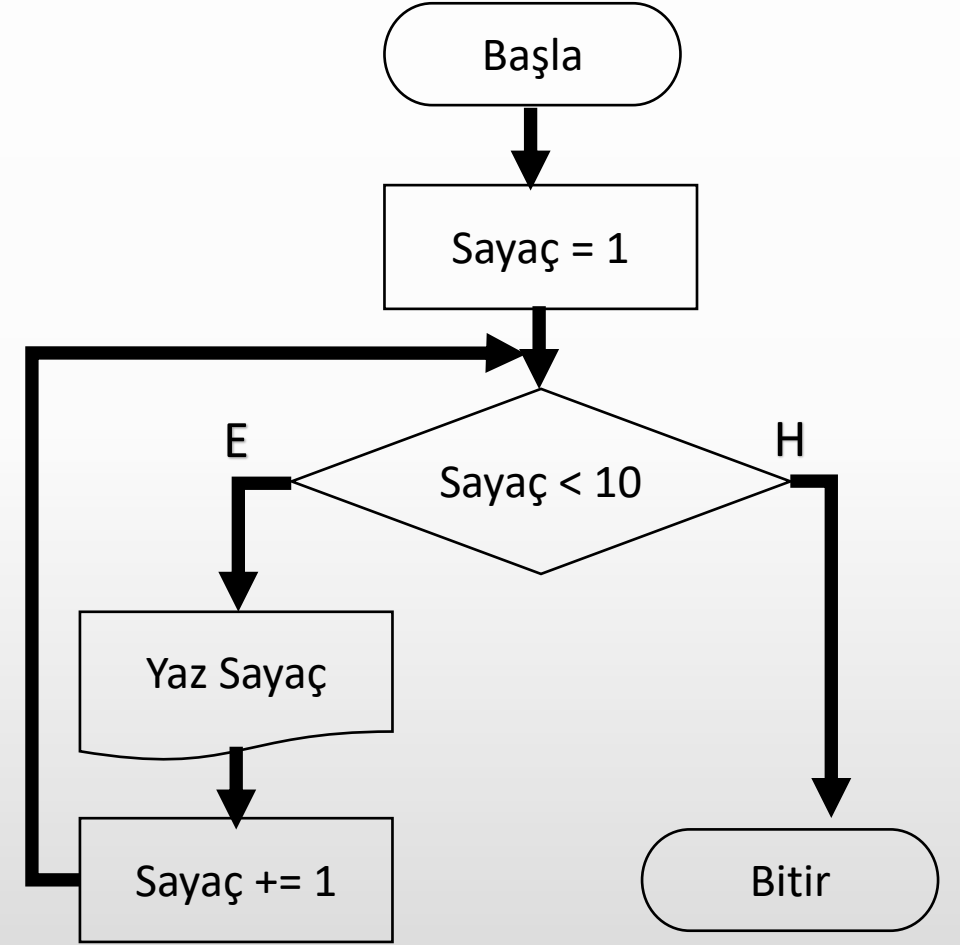


Tekrarlı Akış

- 1'den 10'a kadar olan sayıları yazdırma

- **Algoritma**

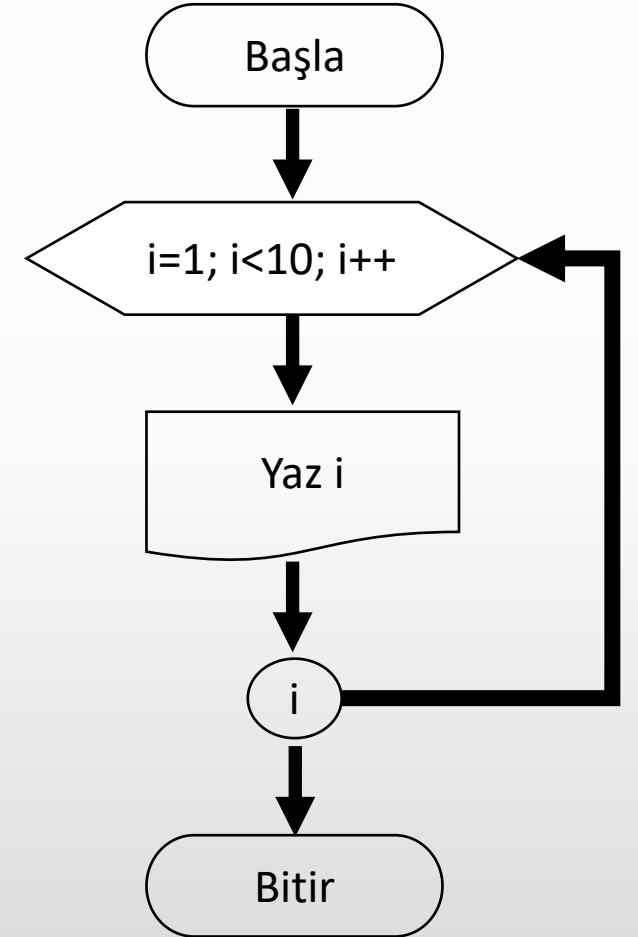
1. Başla
2. Sayaç = 1
3. 10'dan küçükse, 4'e git, değilse 7'ye git
4. Yaz sayaç
5. Sayaç 1 arttır
6. 3'e git
7. Bitir





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