

Bilgisayar Bilimi

ve

Programlama

-4-

Algoritma Nedir?

Akış Diyagramları

Sözde Kod

Programlama Araçları

1. BÖLÜM: ALGORİTMA NEDİR?

1.1. Algoritma Kavramı

Algoritma, bir problemi çözmek veya belirli bir amaca ulaşmak için izlenmesi gereken adımların **sıralı ve mantıksal** bir şekilde ifade edilmesidir. Günlük hayatta farkında olmadan sürekli algoritma uyguluyoruz.

Günlük Hayattan Algoritma Örnekleri:

Örnek 1: Çay Demleme Algoritması

1. Başla
2. Çaydanlığı al
3. Çaydanlığa su doldur
4. Çaydanlığı ocağa koy
5. Çaydanlığın üzerinde durduğu ocağı çalıştır
6. Su kaynayana kadar bekle
7. Demliğe çay koy
8. Kaynayan suyu demliğe dök
9. 5 dakika demlenmesini bekle
10. Çayı bardağa doldur
11. Bitir

Örnek 2: Çamaşır Makinesi Çalıştırma Algoritması

1. Başla
2. Çamaşırları makineye koy
3. Deterjan gözüne deterjan ekle
4. Yumuşatıcı gözüne yumuşatıcı ekle
5. Program seçim düğmesinden "Pamuklu 60°C" seç
6. Başlat düğmesine bas
7. Makine çalışmaya başla
8. Program bitince sinyal ver
9. Çamaşırları makineden çıkar
10. Bitir

1.2. Algoritmanın Temel Özellikleri

Özellik	Açıklama	Örnek
Girdi (Input)	Algoritmanın çalışması için dışarıdan aldığı veriler	Sıcaklık değeri, kuvvet, çap
Çıktı (Output)	Algoritmanın ürettiği sonuç	Hacim, gerilme, karar (Evet/Hayır)
Kesinlik	Her adım net ve anlaşılır olmalı	"Biraz su koy" değil, "200 ml su koy"
Sonluluk	Algoritma sonsuza kadar çalışmamalı, bir yerde bitmeli	10 kez tekrarla, şart sağlanınca dur
Etkililik	Algoritma problemi çözebilmeli	Gerçekten çay demleyebilmeli

1.3. Makine Mühendisliğinde Algoritma Neden Önemlidir?

Makine mühendisliğinde algoritmik düşünce, bir problemi adım adım çözme becerisi kazandırır. Bu beceri şu alanlarda kullanılır:

Mühendislik Alanı	Algoritma Kullanımı
Bilgisayar Destekli Tasarım (CAD)	Bir parçanın oluşturulma sırası
Sonlu Elemanlar Analizi (FEA)	Elemanlara ayırma, matris çözümü
İmalat Süreçleri	İşlem sırası, takım yolu planlaması
Kontrol Sistemleri	PID kontrol algoritması
Isı Transferi	Sıcaklık dağılımının iteratif çözümü

2. BÖLÜM: AKIŞ DİYAGRAMI (FLOWCHART)

2.1. Akış Diyagramı Nedir?

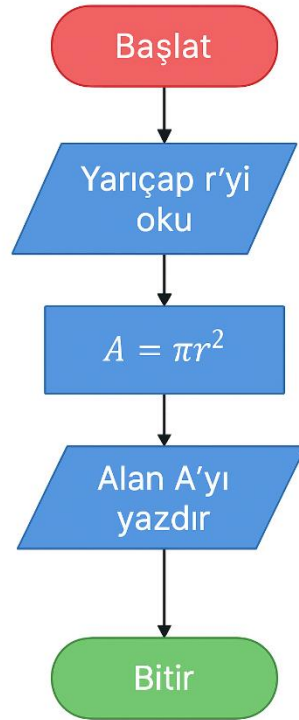
Akış diyagramı, bir algoritmanın **görsel** olarak ifade edilmesidir. Geometrik şekiller ve oklar kullanılarak, işlem adımları ve karar noktaları gösterilir.

2.2. Akış Diyagramı Sembolleri

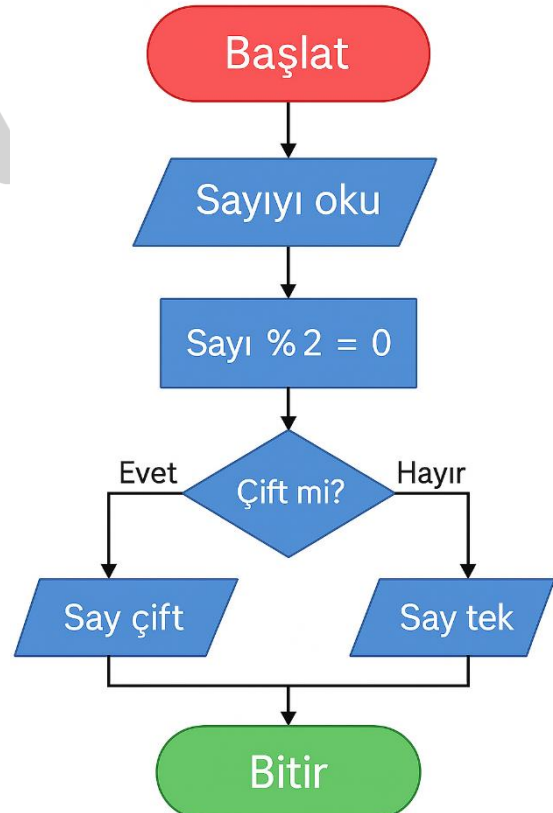
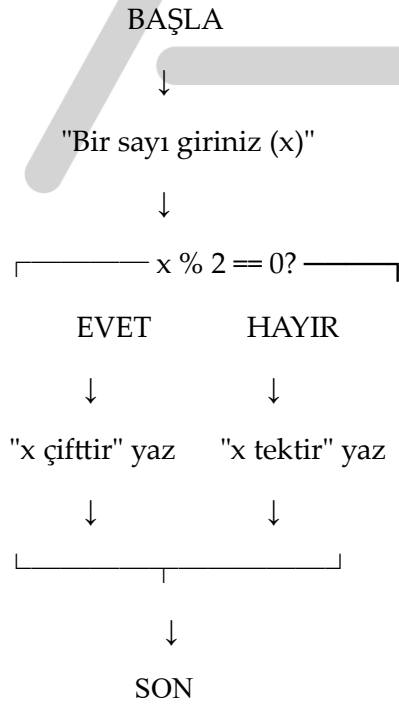
Sembol	Şekil	Anlamı	Mühendislik Örneği
Başla/Bitir		Algoritmanın başlangıcı ve bitişi	Başla, Dur, Son
Girdi/Çıktı		Veri girişi veya sonuç çıkışı	"Çapı gir", "Hacmi yaz"
İşlem		Hesaplama, atama, işlem	$Hacim = \pi \times r^2 \times h$
Karar		Koşul kontrolü, Evet/Hayır	Sıcaklık > 100°C mi?
Döngü		Tekrarlı işlemler	10 kez tekrarla
Akış Yönü		İşlem sırasını gösterir	Adımlararası bağlantı
Bağlantı		Sayfa içi/sayfalar arası bağlantı	A, B, C noktaları

2.3. Akış Diyagramı Örnekleri

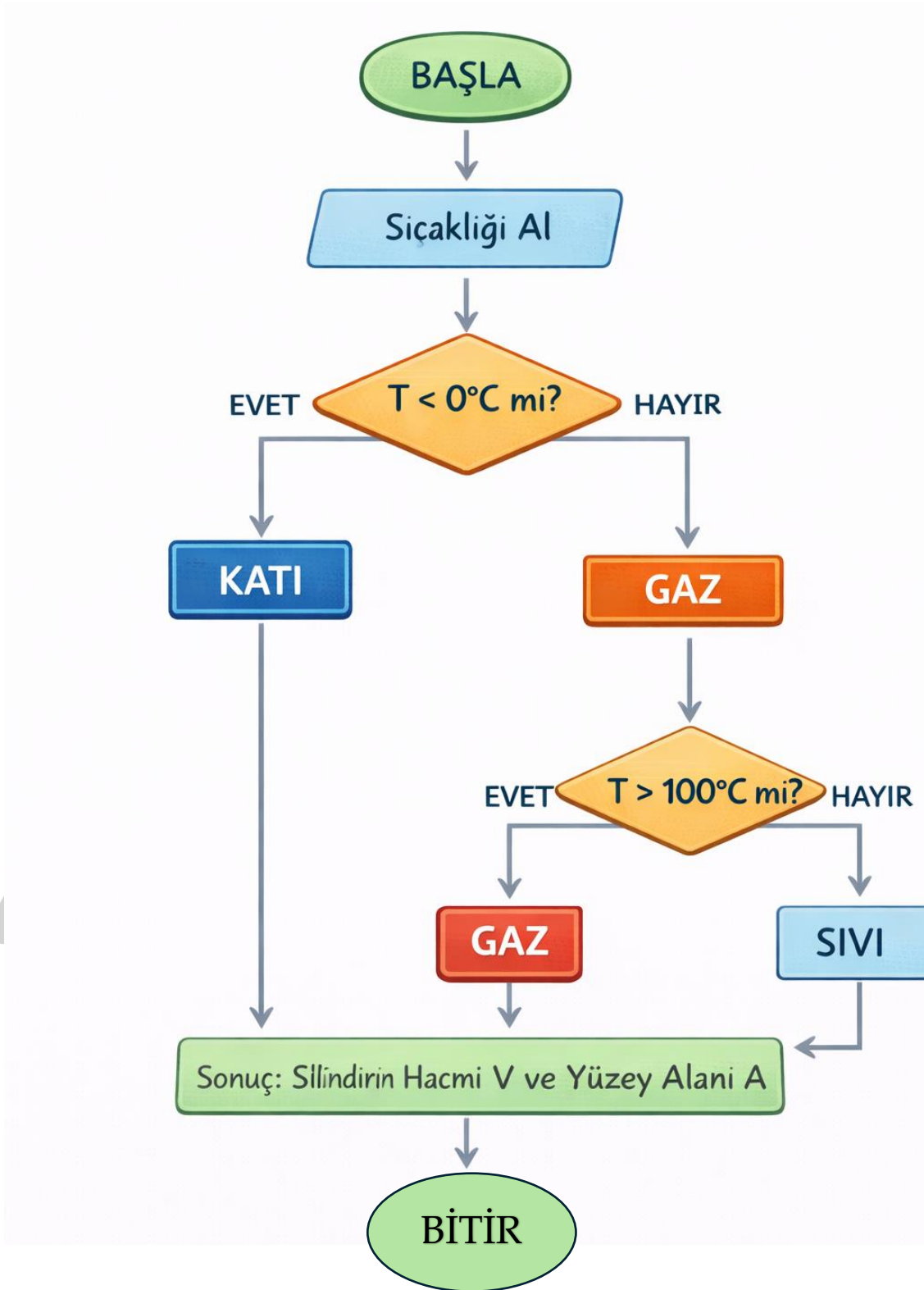
Örnek 1: Daire Alanı Hesaplama



Örnek 2: Sayının Tek mi Çift mi Olduğunu Bulma



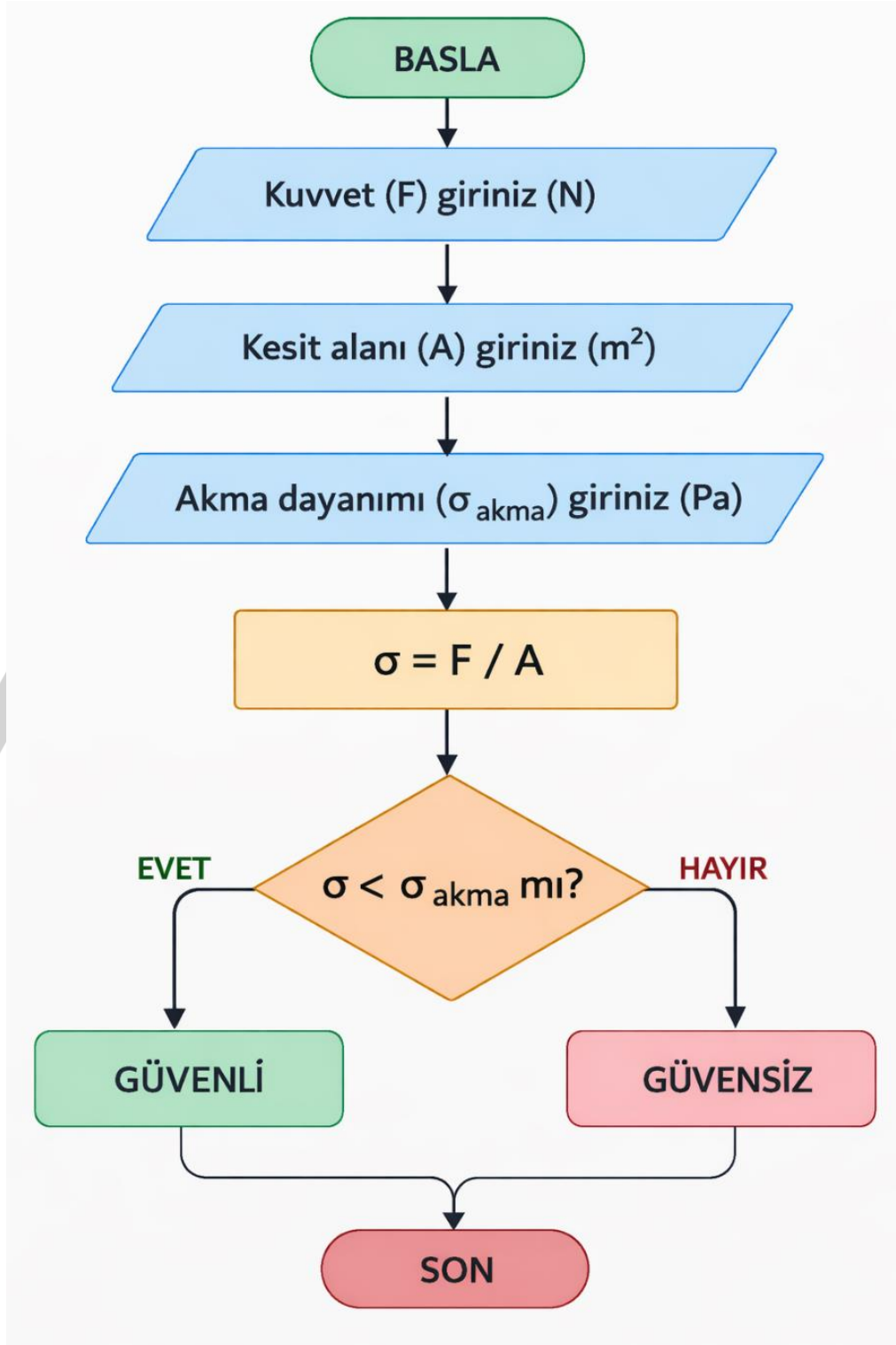
Örnek 3: Sıcaklığa Göre Suyun Fazını Belirleme



2.4. Mühendislik Problemleri için Akış Diyagramları

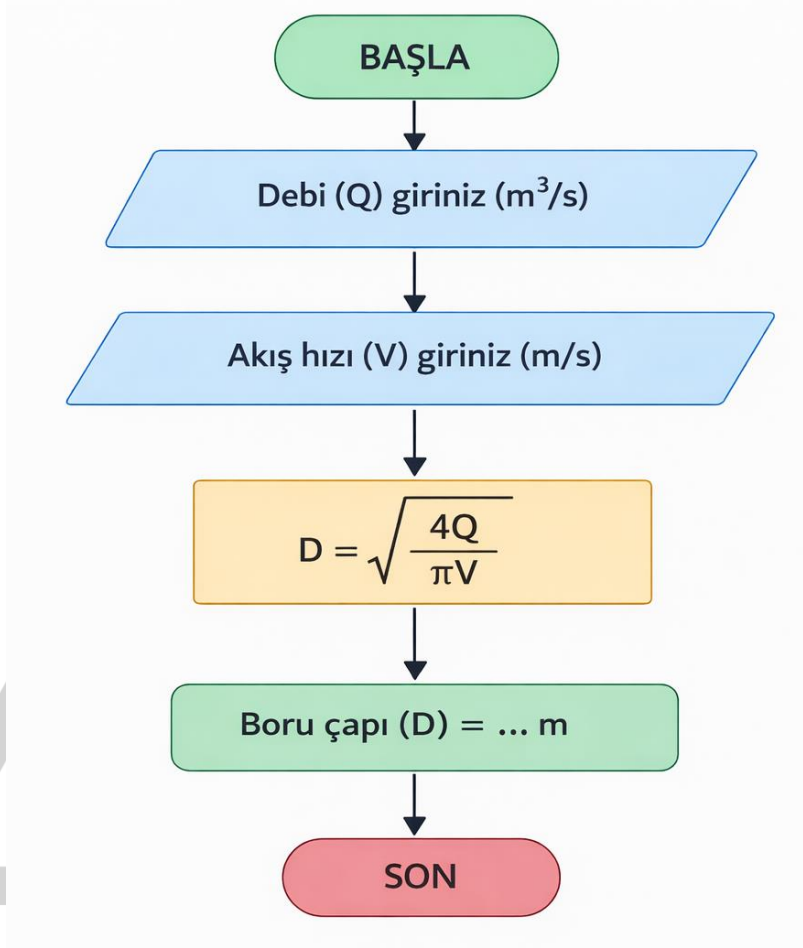
Problem 1: Bir Kirişin Güvenliğini Kontrol Etme

Bir kirişe uygulanan gerilme (σ), malzemenin akma dayanımından (σ_{akma}) küçükse "GÜVENLİ", büyükse "GÜVENSİZ" yazdıran algoritma.



Problem 2: Boru Çapı Seçimi

*Bir boru hattında debi (Q) ve akış hızına (V) göre gerekli boru çapını hesaplayan algoritma.
 $(D = \sqrt{(4Q/\pi V)})^*$

**3. BÖLÜM: SÖZDE KOD (PSEUDO-CODE)****3.1. Sözde Kod Nedir?**

Sözde kod, algoritmanın **doğal dil** (Türkçe/İngilizce) ile programlama dili arasında, **herhangi bir programlama dilinin katı kurallarına bağlı kalmadan** yazılmış halidir. Amaç, mantığı anlaşılır şekilde ifade etmektir.

3.2. Sözde Kod Yazım Kuralları

- Her adım ayrı bir satıra yazılır
- İngilizce veya Türkçe yazılabilir
- Programlama dillerindeki yapılar kullanılabilir (IF, ELSE, FOR, WHILE)
- Girintiler (indent) blokları belirtmek için kullanılır
- { } veya BEGIN-END ile bloklar belirtilebilir

3.3. Sözde Kod Örnekleri

Örnek 1: Dikdörtgen Alanı Hesaplama

ALGORİTMA : Dikdörtgen Alanı Hesaplama

GİRDİ : Kısa kenar (a), Uzun kenar (b)

ÇIKTI : Alan (A)

BAŞLA

// Kullanıcıdan değerleri al

YAZ "Kısa kenarı giriniz: "

OKU a

YAZ "Uzun kenarı giriniz: "

OKU b

// Alanı hesapla

$A = a * b$

// Sonucu göster

YAZ "Dikdörtgenin alanı = " + A

BİTİR

Örnek 2: Sıcaklık Kontrolü (Karar Yapısı)

ALGORİTMA : Sıcaklık Kontrol Sistemi

GİRDİ : Sıcaklık (T)

ÇIKTI : Fan durumu (Açık/Kapalı)

BAŞLA

// Sıcaklık değerini al

YAZ "Oda sıcaklığını giriniz (°C): "

OKU T

// Sıcaklığı kontrol et

EĞER $T > 30$ İSE

YAZ "Fan AÇILIYOR"

fan_durumu = "AÇIK"

DEĞİLSE

YAZ "Fan KAPALI"

fan_durumu = "KAPALI"

EĞER-SONU

YAZ "Fan durumu: " + fan_durumu

BİTİR

Örnek 3: 1'den N'e Kadar Sayıların Toplamı (Döngü)

ALGORİTMA : 1'den N'e Kadar Sayıların Toplamı

GİRİDİ : Üst sınır (N)

ÇIKTI : Toplam (toplam)

BAŞLA

// Başlangıç değerleri

toplam = 0

sayac = 1

// Kullanıcıdan N değerini al

YAZ "Bir N sayısı giriniz: "

OKU N

// 1'den N'e kadar döngü

sayac = 1 İKEN sayac <= N OLDUĞU SÜRECE

toplam = toplam + sayac

sayac = sayac + 1

DÖNGÜ-SONU

// Sonucu göster

YAZ "1'den " + N + "'e kadar sayıların toplamı = " + toplam

BİTİR

4. BÖLÜM: PROGRAMLAMA ARAÇLARI

4.1. Derleyici (Compiler) Nedir?

Derleyici, yazdığınız programı (kaynak kod) bilgisayarın anlayabileceği **makine diline** çeviren programdır. Tüm programı bir kerede çevirir ve çalıştırılabilir dosya (.exe) oluşturur.

Derleyici Çalışma Mantığı: (Örnek: C, C++, Fortran derleyicileri)

Kaynak Kod (C, C++) → DERLEYİCİ → Makine Kodu (.exe) → ÇALIŞTIR

4.2. Yorumlayıcı (Interpreter) Nedir?

Yorumlayıcı, programı **satır satır** okuyarak anında çalıştıran programdır. Ayrı bir çalıştırılabilir dosya oluşturmaz.

Yorumlayıcı Çalışma Mantığı: (Örnek: Python, MATLAB, JavaScript)

Kaynak Kod (Python) → YORUMLAYICI (Her satırı anında çalıştırır)

4.3. Python Neden Yorumlayıcı Kullanır?

Python yorumlayıcı tabanlı bir dildir. Bu sayede:

- Kod yazdıktan hemen sonra sonucu görebilirsiniz
- Hata yaptığınızda anında fark edersiniz
- Mühendislik hesaplamalarında hızlı deneme-yanılma yapabilirsiniz

4.4. IDE (Entegre Geliştirme Ortamı) Nedir?

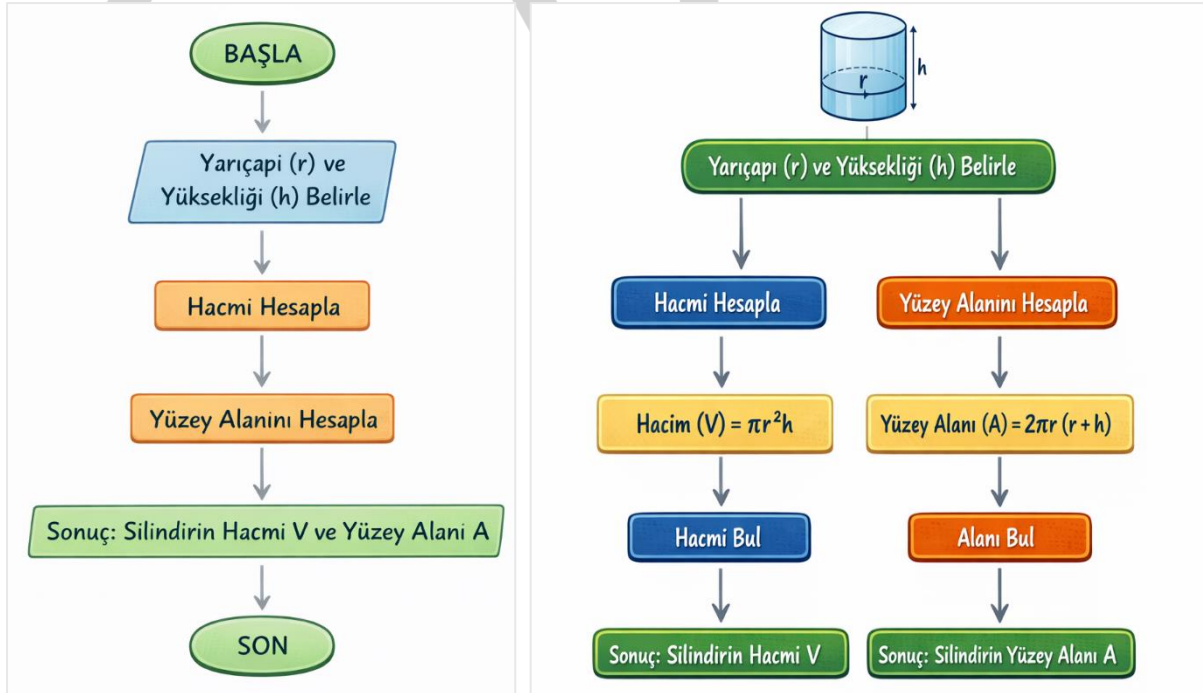
IDE, kod yazmayı kolaylaştıran, içinde **düzenleyici**, **derleyici/yorumlayıcı**, **hata ayıklayıcı** gibi araçları barındıran yazılımdır.

IDE	Özellikler	Kullanım Alanı
PyCharm	Python'a özel, güçlü özellikler	Profesyonel Python geliştirme
Spyder	MATLAB benzeri arayüz	Mühendislik, veri bilimi
VS Code	Hafif, eklenti desteği	Genel amaçlı
Jupyter Notebook	Hücre tabanlı, görselleştirme	Veri analizi, eğitim
IDLE	Python ile gelen basit IDE	Başlangıç seviyesi

5. BÖLÜM: UYGULAMA ÖRNEKLERİ

Örnek 1: Bir Silindirin Hacmini ve Yüzey Alanını Hesaplama

Akış Diyagramı:



a) Tek koldan ilerleyen akış diyagramı b) Paralel hesaplama yapılan (2 koldan) akış diyagramı

Sözde Kod:

ALGORİTMA : Silindir Hacim ve Yüzey Alanı

GİRDİ : Yarıçap (r), Yükseklik (h)

ÇIKTI : Hacim (V), Yüzey Alanı (A)

BAŞLA

// Sabit tanımı

$\pi = 3.14159$

// Kullanıcıdan değerleri al

YAZ "Silindir yarıçapını giriniz: "

OKU r

YAZ "Silindir yüksekliğini giriniz: "

OKU h

// Hesaplamalar

$V = \pi \times r \times r \times h$

$A = 2 \times \pi \times r \times (r + h)$

// Sonuçları göster

YAZ "Silindir Hacmi: " + V

YAZ "Silindir Yüzey Alanı: " + A

BİTİR

Örnek 2: Bir Kirişteki Gerilme ve Güvenlik Faktörü Hesaplama

Problem: Bir kirişe uygulanan çekme kuvveti (F) ve kesit alanına (A) göre gerilmeyi (σ) hesaplayın. Malzemenin akma dayanımı (σ_{akma}) biliniyorsa güvenlik faktörünü (S) bulun.

Sözde Kod:

ALGORİTMA : Gerilme ve Güvenlik Faktörü Hesabı

GİRDİ : Kuvvet (F), Kesit alanı (A), Akma dayanımı (σ_{akma})

ÇIKTI : Gerilme (σ), Güvenlik faktörü (S)

BAŞLA

// Kullanıcıdan değerleri al

YAZ "Uygulanan kuvveti (N) giriniz: "

OKU F

YAZ "Kesit alanını (m²) giriniz: "

OKU A

YAZ "Malzemenin akma dayanımını (Pa) giriniz: "

OKU σ_{akma}

```
// Gerilmeyi hesapla
 $\sigma = F / A$ 

// Güvenlik faktörünü hesapla
 $S = \sigma_{akma} / \sigma$ 

// Sonuçları göster
YAZ "Hesaplanan Gerilme: " +  $\sigma$  + " Pa"
YAZ "Güvenlik Faktörü: " + S

// Güvenlik durumunu kontrol et
EĞER  $S \geq 2$  İSE
    YAZ "Durum: GÜVENLİ ( $S \geq 2$ )"
DEĞİLSE EĞER  $S \geq 1.5$  İSE
    YAZ "Durum: SINIRDA ( $1.5 \leq S < 2$ )"
DEĞİLSE
    YAZ "Durum: GÜVENSİZ ( $S < 1.5$ )"
EĞER-SONU
BİTİR
```

Örnek 3: Sıcaklık Birim Dönüşümü ($^{\circ}\text{C} \leftrightarrow ^{\circ}\text{F}$)

Sözde Kod:

ALGORİTMA : Sıcaklık Birim Dönüşümü

GİRDİ : Seçim, Sıcaklık değeri

ÇIKTI : Dönüştürülmüş sıcaklık

BAŞLA

```
// Dönüşüm tipini seç
YAZ "Sıcaklık Dönüşüm Programı"
YAZ "1. Celsius → Fahrenheit"
YAZ "2. Fahrenheit → Celsius"
YAZ "Seçiminiz (1 veya 2): "
OKU secim

// Seçime göre işlem yap
EĞER secim = 1 İSE
    YAZ "Celsius değerini giriniz: "
    OKU C
     $F = (C \times 9/5) + 32$ 
    YAZ C + " °C = " + F + " °F"

DEĞİLSE EĞER secim = 2 İSE
```

YAZ "Fahrenheit değerini giriniz: "

OKU F

$$C = (F - 32) \times 5/9$$

YAZ F + " °F = " + C + " °C"

DEĞİLSE

YAZ "Geçersiz seçim! Lütfen 1 veya 2 giriniz."

EĞER-SONU

BİTİR

Örnek 4: Motor Devrine Göre Vites Değiştirme Sistemi

Problem: Bir araçta motor devri (RPM) 2000'in altındaysa vites düşür, 3000'in üzerindeyse vites yükselt, arada ise vitesi koru.

Sözde Kod:

ALGORİTMA : Vites Değiştirme Sistemi

GİRDİ : Motor devri (rpm)

ÇIKTI : Vites değiştirme önerisi

BAŞLA

// Motor devrini al

YAZ "Anlık motor devrini (rpm) giriniz: "

OKU rpm

// Devre göre vites öner

EĞER rpm < 2000 İSE

YAZ "VİTES DÜŞÜRÜN (Düşük devir)"

DEĞİLSE EĞER rpm > 3000 İSE

YAZ "VİTES YÜKSELTİN (Yüksek devir)"

DEĞİLSE

YAZ "VİTESİ KORUYUN (Optimum devir)"

EĞER-SONU

// Ek bilgi

EĞER rpm < 1000 İSE

YAZ "UYARI: Motor stop edebilir!"

EĞER rpm > 4000 İSE

YAZ "UYARI: Motor aşırı devir!"

BİTİR

6. BÖLÜM: ALIŞTIRMALAR

Alıştırma 1: Dikdörtgen Kesitli Kirişin Atalet Momenti

Dikdörtgen kesitli bir kirişin atalet momenti $I = (b \times h^3) / 12$ formülü ile hesaplanır. b (taban genişliği) ve h (yükseklik) değerlerini kullanıcından alan, atalet momentini hesaplayan algoritmanın:

- Akış diyagramını çiziniz
- Sözde kodunu yazınız

Alıştırma 2: Isıtıcı Kontrol Sistemi

Bir ısıtıcı, sıcaklık 20°C'nin altında ise çalışır, 25°C'nin üzerinde ise durur. 20-25°C arasında ise mevcut durumunu korur. Bu sistemi akış diyagramı ile gösteriniz.

Alıştırma 3: Pompa Seçimi

Bir pompa seçimi için gerekli debi (Q) ve basınç (P) değerlerine göre:

- $Q < 10 \text{ m}^3/\text{s}$ ve $P < 5 \text{ bar}$ ise "Küçük pompa"
- $Q < 50 \text{ m}^3/\text{s}$ ve $P < 10 \text{ bar}$ ise "Orta pompa"
- Diğer durumlarda "Büyük pompa" yazdıran algoritmayı sözde kod ile yazınız.

Alıştırma 4: Faktöriyel Hesaplama

Bir sayının faktöriyelini (n!) hesaplayan algoritmanın akış diyagramını çiziniz. (Örnek: $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$)

Alıştırma 5: Maksimum Gerilme Bulma

Bir deneyde ölçülen 5 farklı gerilme değeri arasından en büyük olanı bulan algoritmanın sözde kodunu yazınız.