



C PROGRAMLAMA EĞİTİMİ

Gazi Üniversitesi Yapay Zeka
Topluluğu

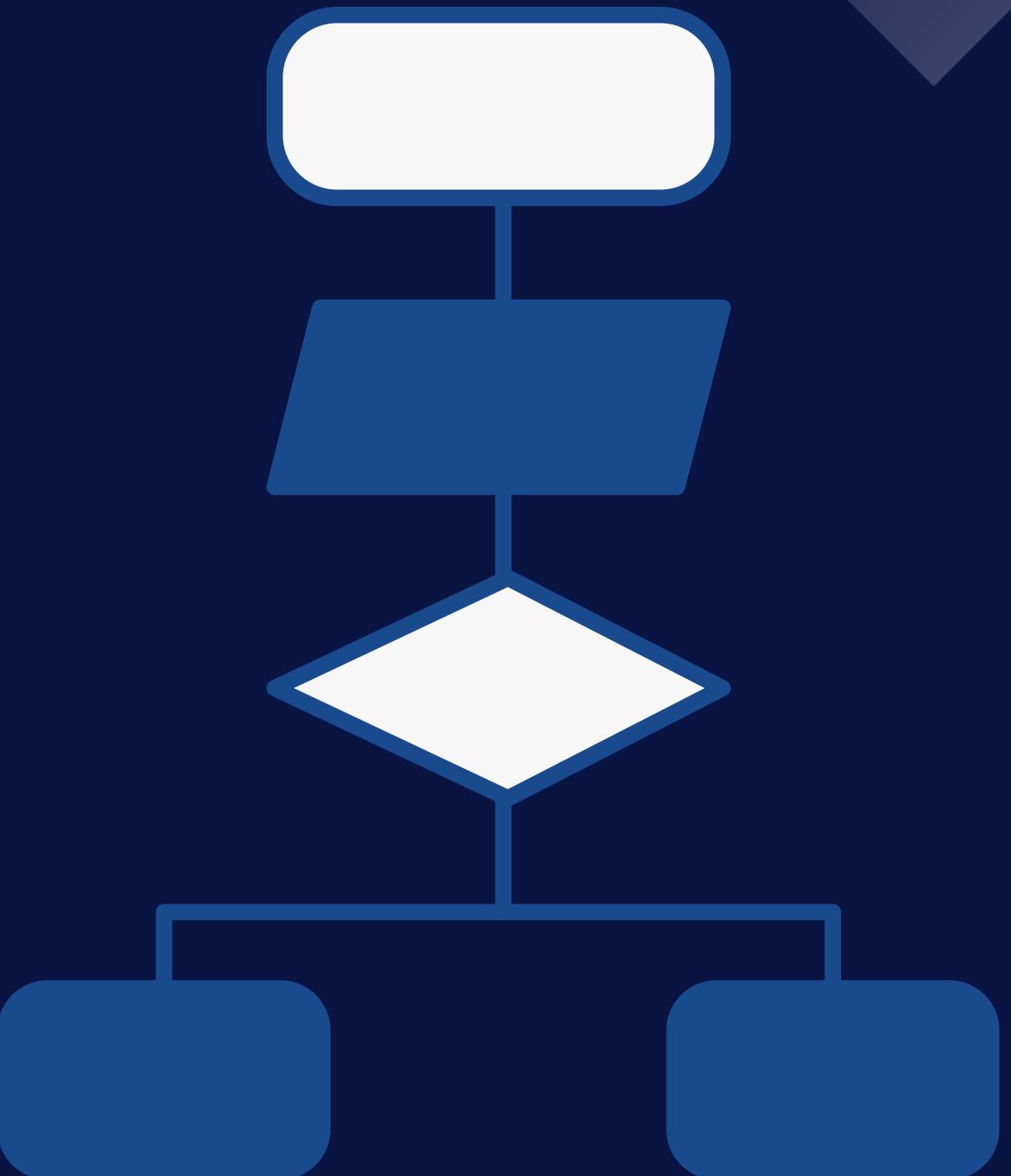
KONU BAŞLIKLERİ

- Algoritmalar
- C Diline Giriş
- Veri Tipleri
- Operatörler
- Karar Yapıları
- Döngü Yapıları
- Fonksiyonlar



Algoritmalar ve C Diline Giriş

- Algoritma Nedir?
- C Diline Giriş
- C Temelleri



Algoritma Nedir

- Belirli bir problemi çözmek için izlenen sonlu adımlar dizisidir.
- Bilgisayar programlarının temelini oluşturur.
- Akış diyagramları ile görselleştirilebilir.



AKİŞ DİYAGRAMLARI

- Algoritmanın görsel gösterimidir.
- Başlangıç, karar, işlem ve bitiş sembollerini kullanılır.
- Program mantığını anlamayı kolaylaştırır.

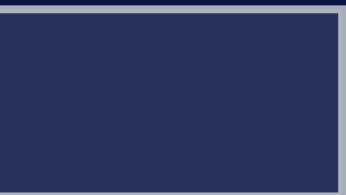




Başla / Bitir



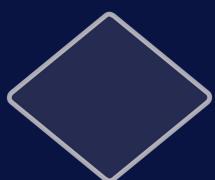
Bilgi Giriş Çıkışı



İşlem



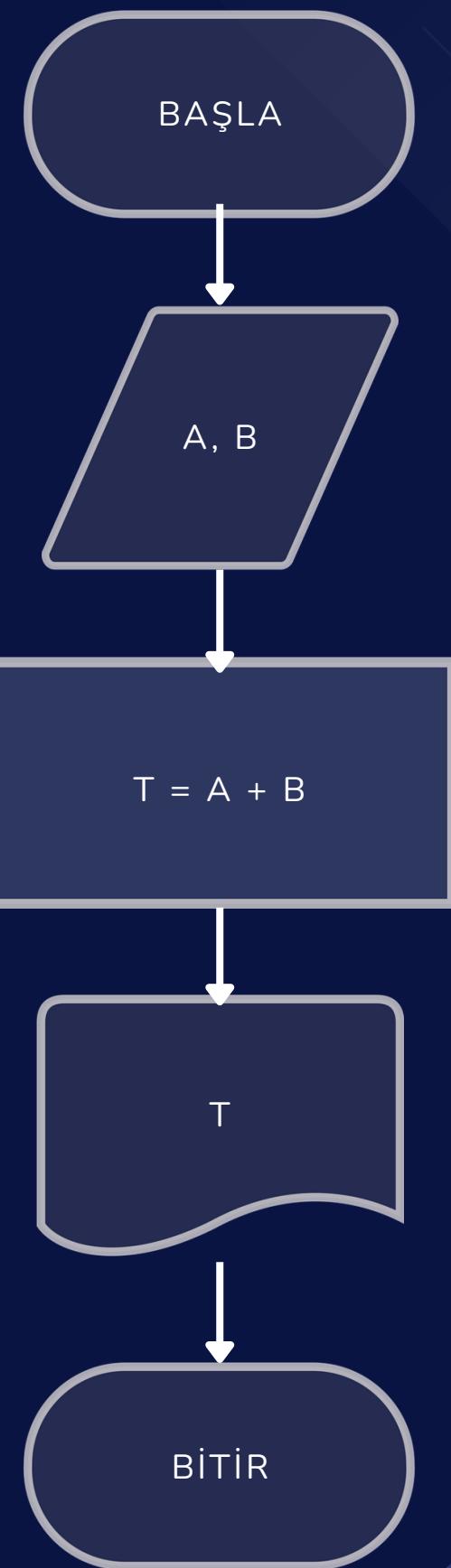
Ekrana Çıktı Gösterme



Karar Yapısı



Birleştirme Okları



Programlama

- Programlama dilleri, bilgisayarların anlayacağı komutları daha kolay ifade etmemizi sağlar.
- Makine dili: 0 ve 1'lerden oluşur, en düşük seviyedir. İnsanlar için yazması/anlaması zordur.
- Yüksek seviyeli diller: İnsan diline daha yakın sözdizimine sahiptir (C, Java, vb.).



C Diline Giriş

- Sistem programlama için geliştirilmiş güçlü bir dildir.
- Derlenir, yani kaynak kod makine diline çevrilir.
- Farklı işletim sistemlerinde çalışabilir.



C Dilinin Tarihçesi

- 1970'lerin başında Bell Labs'ta Dennis Ritchie tarafından geliştirildi.
- 1972: UNIX işletim sistemi büyük ölçüde C ile yazıldı → dilin popülerliği hızla arttı.
- 1989: ANSI C standardı kabul edildi → farklı derleyicilerde uyumluluk sağlandı.
- 1990 ve sonrası: ISO standartlarıyla güncellendi, modern C versiyonları (C99, C11, C17) ortaya çıktı.



C Dilinin Özellikleri

- Hızlı ve verimli: Derlenmiş programlar makine diline çok yakın çalışır.
- Taşınabilir: Aynı C kodu farklı işletim sistemlerinde küçük değişikliklerle çalışabilir.
- Modüler yapı: Fonksiyonlar ve kütüphanelerle büyük projeler parçalanabilir.



C Dilinin Kullanım Alanları

- İşletim sistemleri: UNIX, Linux çekirdeği, Windows'un bazı bölümleri C ile yazılmıştır.
- Gömülü sistemler: Mikrokontrolcüler, IoT cihazları, otomotiv elektroniği.
- Sistem yazılımları: Derleyiciler, sürücüler, ağ protokollerı.
- Diğer dillere temel: C++, Objective-C, C#, Java gibi diller C'nin sözdizimi ve mantığından etkilenmiştir.



Neden C?

- Temel programlama mantığını ve bilgisayarın nasıl çalıştığını anlamaya yardımcı olur.
- Donanım ve yazılım arasındaki ilişkiyi görmemizi sağlar.
- Daha karmaşık dillere geçiş için sağlam bir temel oluşturur.



İlk Programımızı Yazalım



Veri Tipleri

- Veri tipleri programlarımıza farklı tip verileri temsil etmemizi sağlar.
- Gerçek hayatta bilgilerin farklı ölçülerle ifade edildiği gibi programlamada da veriler farklı tiplerle ifade edilir.



| Tip | Boyut | Aralık | Örnek |
|-------|-------|----------------------------------|--------------|
| char | 1 | -128, 127 | 1, 'a' |
| short | 2 | -32.768, 32.767 | 132 |
| int | 4 | -2.147.483.648, 2.147.483.647 | 5, 23, -4389 |



| Tip | Boyut (byte) | Aralık | Örnek |
|--------|--------------|----------------|----------------|
| long | 8 | çok büyük | 223.234.634... |
| float | 4 | $3.4E +/- 38$ | 3.1425 |
| double | 8 | $1.7E +/- 308$ | 42.23423... |



Değişkenler

- Değişkenler program içinde verileri tutmak için kullandığımız yapılardır.
- Aynı verinin birden fazla yerde kullanılması gereken durumlarda değişken kullanmak oldukça mantıklıdır.



```
int sayıl = 12;
```

Tür
Belirteci

Değişken
İsmi

Değer



Veri Tiplerini ve Değişkenleri Kod İçinde Kullanalım



Operatörler

- Programlamada operatörler matematiksel ve mantıksal işlemler için kullanılır.



| Operatör | Görev |
|----------|---------------|
| + | Toplama |
| - | Çıkarma |
| * | Çarpma |
| / | Bölme |
| % | Modüler Bölme |



| Operatör | Görev |
|----------|-----------------------|
| = | Atama Operatörü |
| += | Toplayarak Atama |
| -= | Çıkararak Atama |
| *= | Çarparak Atama |
| /= | Bölerek Atama |
| %= | Modüler Bölerek Atama |



| Operatör | Görev |
|--------------------|-----------------------------------|
| <code>==</code> | Eşitlik Kontrolü |
| <code>!=</code> | Eşitsizlik Kontrolü |
| <code><</code> | Küçüklük Karşılaştırma |
| <code>></code> | Büyüklük Karşılaştırma |
| <code><=</code> | Küçüklük ve eşitlik karşılaştırma |
| <code>>=</code> | Büyüklük ve eşitlik karşılaştırma |



| Operatör | Görev |
|----------|--------------------|
| && | Mantıksal Ve |
| | Mantıksal Veya |
| ++ | Artırma Operatörü |
| -- | Eksiltme Operatörü |



Operatörleri Kod İçinde Kullanım



Dinlediğiniz İçin Teşekkürler

 @gaziyahapayzeka

 Gazi Üniversitesi Yapay Zekâ
Topluluğu

 Ramazan Karataş

 @Orama_k

