

# Programlama Dillerinin Prensipleri

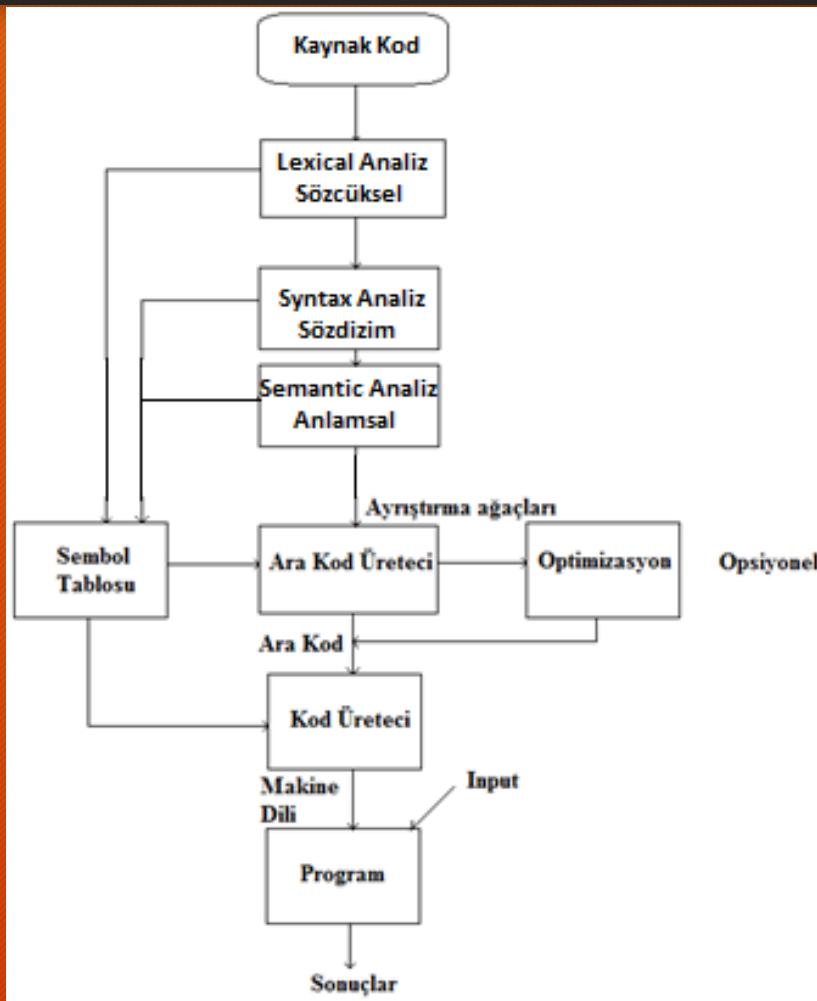
## Hafta 3 - Dillerin Çevrimi Tanımlanması ve Tasarımı

Dr. Öğr. Üyesi M. Fatih ADAK

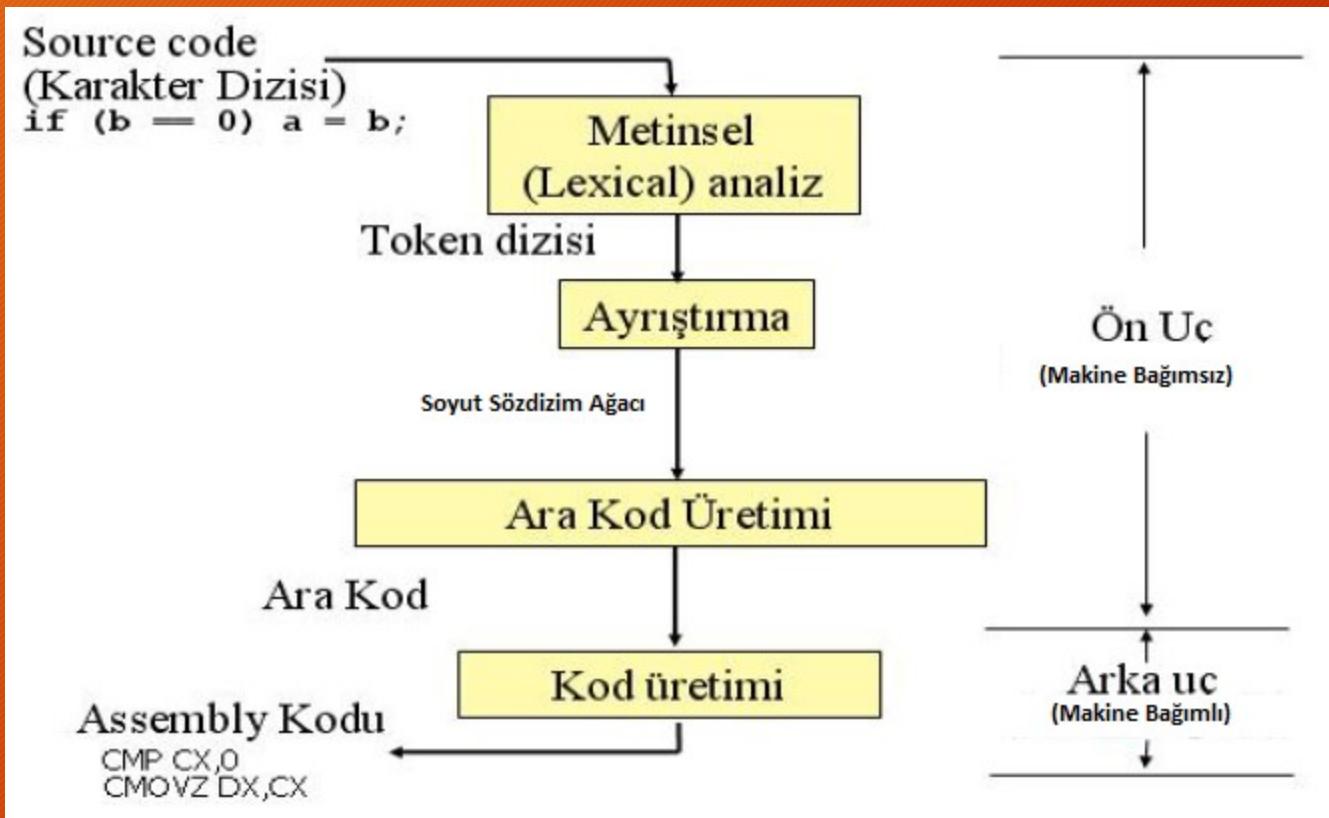
# İçerik

- Derlenme temel süreçler
- Lexical analiz
- Sözdizim analizi
- Dilbigisi kuralları
- BNF - EBNF
- Anlamsal analiz
- Kod Optimizasyonu
- Ara Kod Üreteci
- Kod Üretimi
- Derleyici Yorumlayıcı

# Derlenme Temel Süreçler



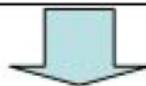
# Derlenme Temel Süreçler



# Sözcüksel (Lexical) Analiz

- Kaynak programın en alt düzeyli birimlerini (lexeme) belirler ve ayırt eder.
- Tek başına bir anlam taşıyan karakter katarlarını tanır ve bunları ayırır.

```
program gcd (input, output);
var i, j : integer;
begin
  read (i, j);
  while i <> j do
    if i > j then i := i - j else j := j - i;
    writeln (i)
end.
```

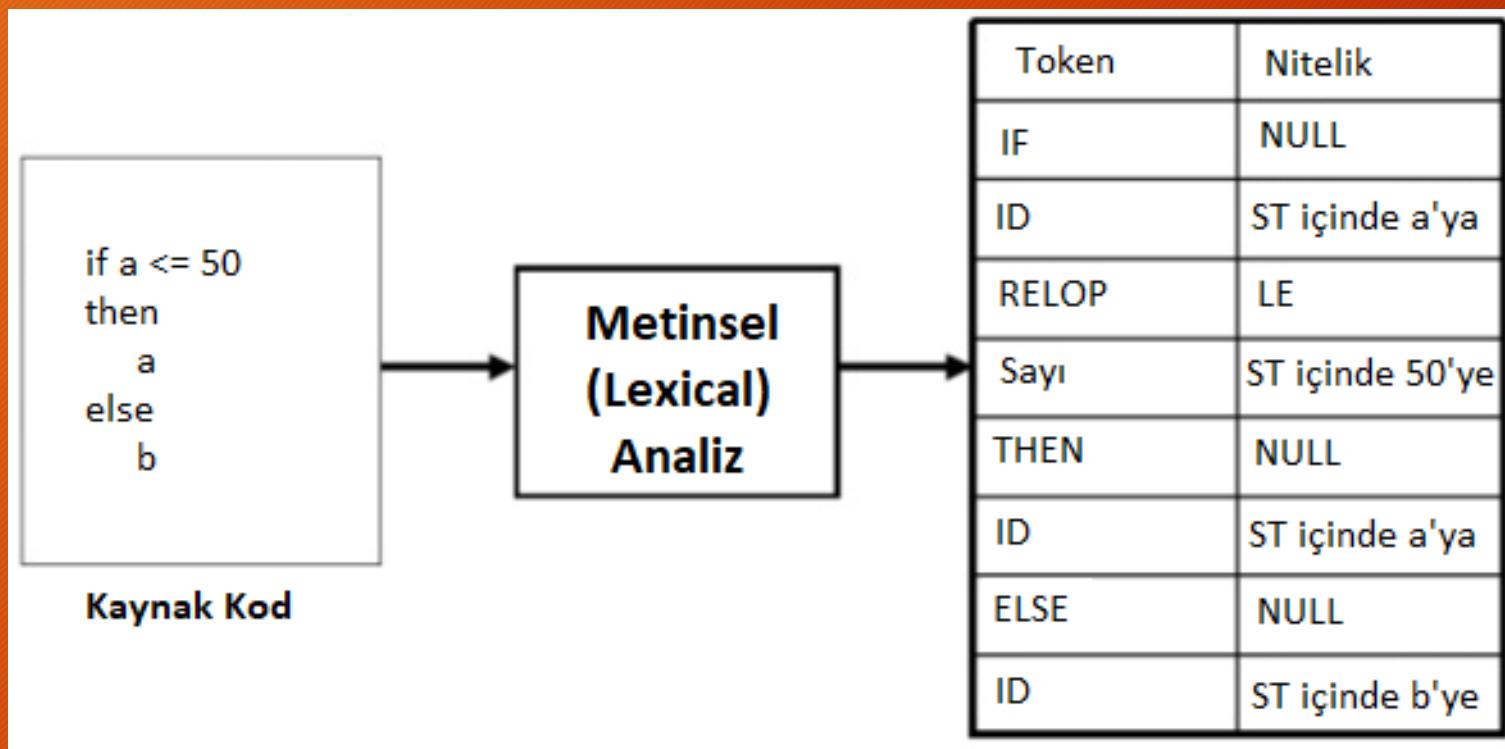


```
program      gcd (      input ,      output ) ; begin
var          i ,      j :      integer ; begin
read         (      i ,      j ) ;      while
i           <> j do      if      i
then        i := i - j      else      j
:=        i - i ;      writeln (      i
)      end .
```

# Token Listesinin Oluşumu

Sonuc := veriler / 25		
Sonuc	(degisken,1)	tanımlayıcı
:=	(atama, nil)	Atama İşlemcisi
veriler	(değişken, 2)	tanımlayıcı
/	(bolme, nil)	Bölme İşlemcisi
25	(tamsayı, 3)	tamsayı

# Token Listesinin Oluşumu



# Sözdizim Analizi

- Kaynak programı oluşturan sözcüklerin programlama dilinin gramer kurallarına uygun bir sıralamada olup olmadıklarını belirler.
- Bu işlem için dilin gramer kurallarını ifade eden BNF ve CFG gibi araçlardan yararlanır.
- Eğer token dizisi için gramer bir parse ağacı oluşturuyorsa ifade söz dizim kuralları yönüyle doğru yazılmıştır ve gramer katarı üreten türetim adımlarını verir.

# Dilbilgisi (Gramer)

- Dilbilgisi, bir programlama dilinin metinsel (somut) sözdizimini açıklamak için kullanılan bir gösterimdir.
- BNF (Backus - Naur Formu) ve CFG (Context-Free Gramerler) gibi gramerler somut söz dizimini açıklamak için geliştirilmiş gösterim araçlarıdır.
- CFG içerikten bağımsız dilbilgisi gösterim şeklidir. Belirsizlik içeren du-rumlar CFG ile gösterilemez.

# Grameri Oluşturan Yapılar

- Başlangıç nonterminali
- Nontermineller
- Terminaller
- Kurallar

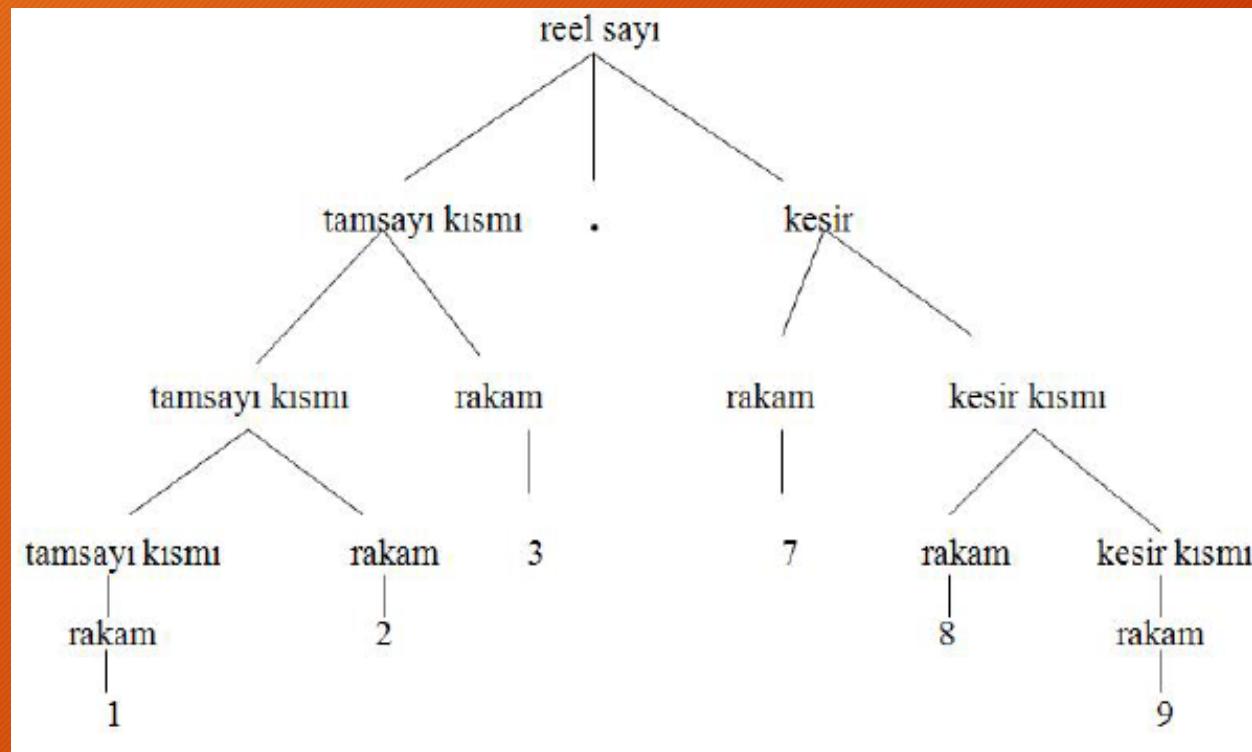
# BNF

## Reel sayılar için BNF kuralları

```
<reel sayı>      ::=  <tam sayı - kısım> . <kesir>
<tam sayı - kısım> ::=  <sayı> | <tam sayı - kısım> <sayı>
<kesir>          ::=  <sayı> | <sayı> <kesir>
<sayı>           ::=  <rakam> | <rakam>
<rakam>          ::=  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<kesir>          ::=  <rakam> | <rakam> <kesir>
```

# Bir Kuralın Ayrıştırma Ağacı ile ifade edilmesi

## Reel sayılar için Ayrıştırma Ağacı



# Belirsizlik

- Bir dilin grameri eğer bazı katarlar için birden çok parse ağacı oluşturabiliyorsa bu gramer belirsiz bir gramerdir.
- Belirsizliği ortadan kaldırmak için belirsizliğinoluştuğu yere yeni non-terminal simgeler eklenir ve sadece sol-rekürsif veya sağ-rekürsif yapıya izin verilir.

$$E ::= E - E \mid 0 \mid 1$$

# Sağ ve Sol Rekürsif

Sol-rekürsif CFG'ye örnek

$$S \rightarrow Ra \mid a$$
$$R \rightarrow ab \mid Rb$$

Sağ-rekürsif CFG'ye örnek

$$S \rightarrow aR \mid a$$
$$R \rightarrow ab \mid bR$$

# EBNF

- BNF gösteriminin genişletilmiş halidir.
- Yineleme (repetition), Seçimlik (optionality) ve Değiştirme (alternation) olmak üzere üç özellik yer almaktadır.
- EBNF'nin avantajı gösterimi kolaylaştırmasıdır.

# EBNF Özellikleri

- Yineleme

EBNF	BNF
<code>&lt;fade_listesi&gt; ::= {&lt;fade&gt;;}</code>	<code>&lt;fade_listesi&gt; ::= &lt;boş&gt;</code>
	<code>  &lt;fade&gt; ; &lt;fade_listesi&gt;</code>

- Değiştirme

EBNF	BNF
<code>&lt;id&gt; ::= &lt;karakter&gt;</code>	
	<code>  &lt;id&gt; &lt;karakter&gt;</code>
	<code>  &lt;id&gt; &lt;rakam&gt;</code>

EBNF
<code>&lt;id&gt; ::= &lt;karakter&gt; (&lt;karakter&gt;   &lt;rakam&gt;)*</code>

- Seçimlik

<b>EBNF</b>	<code>&lt;gerçel sayılar&gt; ::= [tam kısım].&lt;kesirli ifade&gt;</code>
<b>BNF</b>	<code>&lt;gerçel sayı&gt; ::= &lt;tam kısım&gt; . &lt;kesirli ifade&gt;   . &lt;kesirli ifade&gt;</code>

# BNF - EBNF

## BNF

```
<fade> ::= <fade> + <terim>
        | <fade> - <terim>
        | <terim>
<terim> ::= <terim> * <faktör>
        | <terim> / <faktör>
        | <faktör>
<faktör> ::= sayı | isim | (<fade>)
```

## EBNF

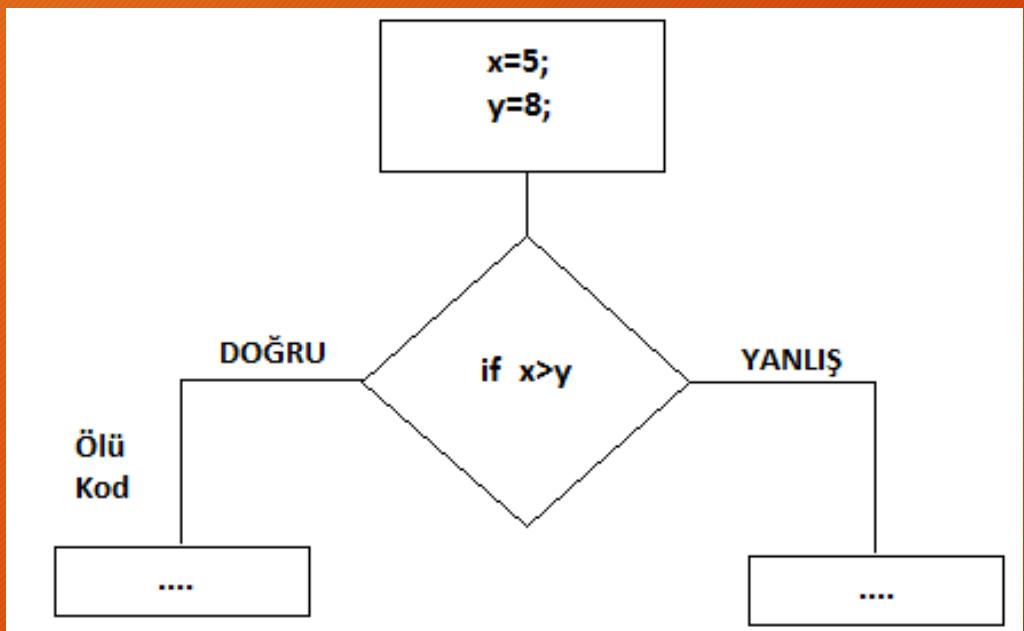
```
<fade> ::= <terim>{(+|-)<terim>}
<terim> ::= <faktör>{(*|/)<faktör>}
<faktör> ::= ('<fade>') | isim | sayı
```

# Anlamsal Semantik Analiz

- Gramer sadece dilin sözdizimini tanımlayabilir. Dilin çevrilmesi sırasında gerek duyulan bazı bilgileri tanımlamada regüler ifade ya da gramer kullanılamaz.
- Bu durumlarda semantik analiz devreye girer.
  - Tür atamaları
  - Null nesne erişimi
  - Değer atamadan okuma

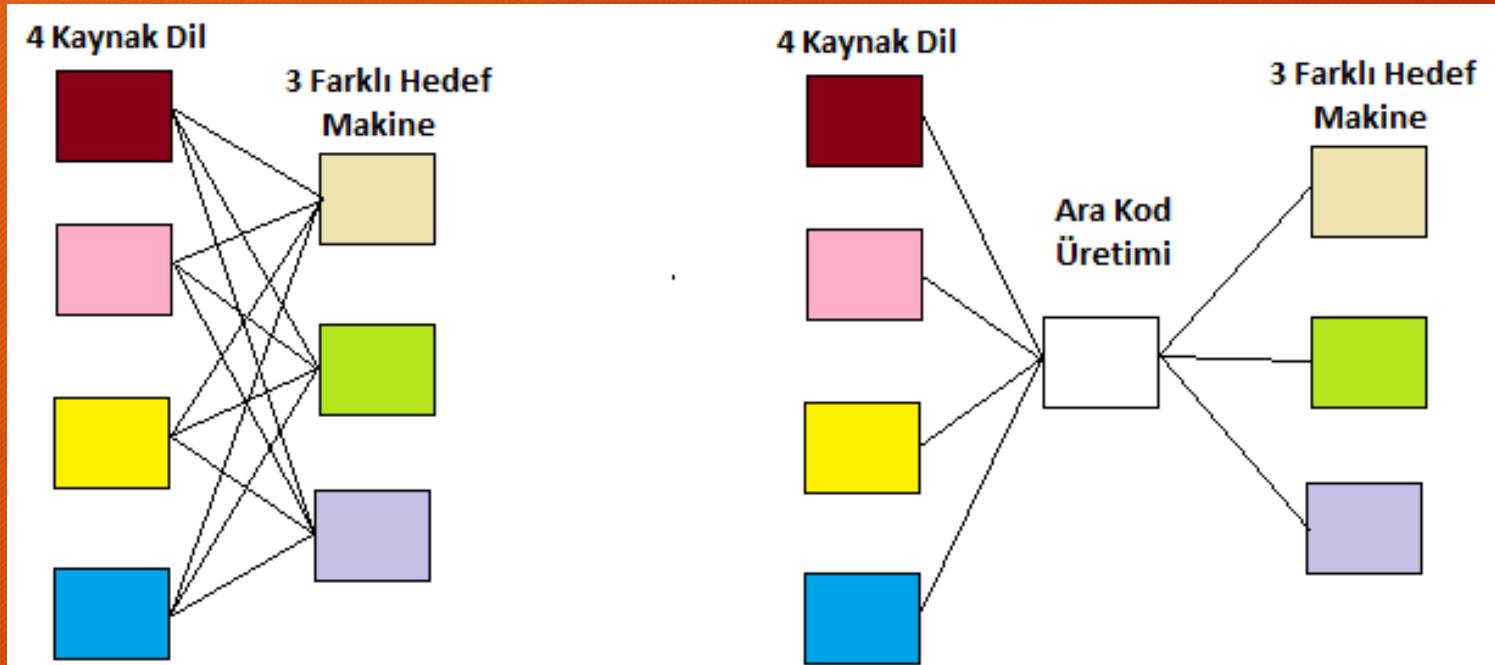
# Kod Optimizasyonu - Opsiyonel

- Üretilen kodun, olası çalışma zamanını ya da bu kodu saklamak için gerekli bellek alanını azaltmak için yapılacak iyileştirmeler sürecidir.



<code>do{ deger = 10; toplam = toplam + deger; } while(toplam&lt;100);</code>	<code>deger=10; do{ toplam = toplam + deger; } while(toplam&lt;100);</code>
---	---

# Ara Kod Üreteci



# Üçlü Adres Formu

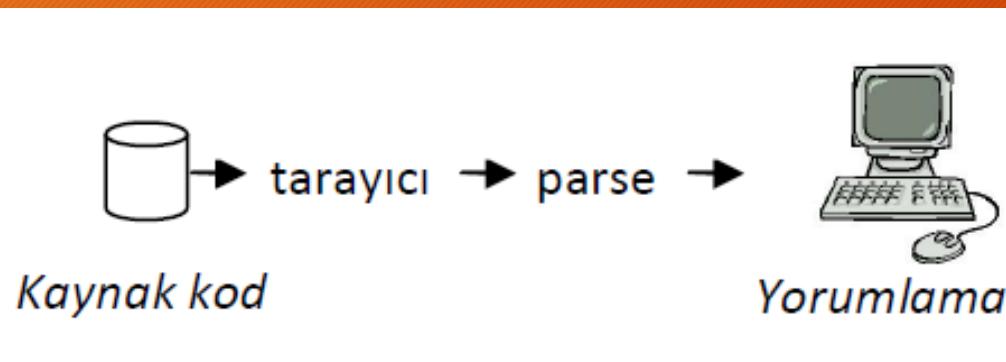
$T3=T1*T2;$

# Kod Üretimi

- Bu aşamada donanımdan bağımsız ara koddan hedef donanımın simgesel/makina koduna dönüşüm (Sanal makinadan fiziksel makinaya geçiş) gerçekleştirilir.

# Derleyici ve Yorumlayıcı

Yorumlayıcı



Derleyici



# Kaynaklar

- Yumusak N., Adak M.F. *Programlama Dillerinin Prensipleri*. 1. Baskı, Seçkin Yayıncılık, 2018
- Sebesta, Robert W. *Concepts of programming languages*. 11 ed. Pearson Education Limited, 2016.
- Sethi, Ravi. *Programming languages: concepts and constructs*. Addison Wesley Longman Publishing Co., Inc., 1996.
- Watt, David A. *Programming language design concepts*. John Wiley & Sons, 2004.
- Malik, D. S., and Robert Burton. *Java programming: guided learning with early objects*. Course Technology Press, 2008.
- Waite, Mitchell, Stephen Prata, and Donald Martin. *C primer plus*. Sams, 1987.
- Hennessey, Wade L. *Common Lisp*. McGraw-Hill, Inc., 1989.
- Liang, Y. Daniel. *Introduction to Java programming: brief version*. pearson prentice hall, 2009.
- Yumusak N., Adak M.F. *C/C++ ile Veri Yapıları ve Çözümlü Uygulamalar*. 2. Baskı, Seçkin Yayıncılık, 2016