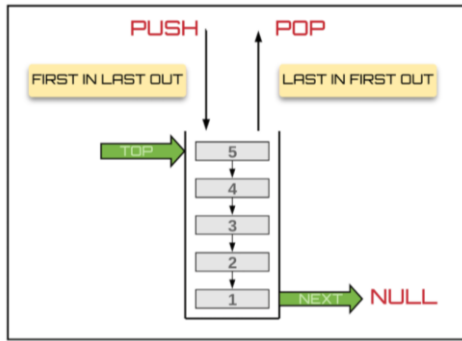


## INFIX – PREFIX – POSTFIX

### YIĞINLAR

Stack, doğrusal artan bir veri yapısı olup; insert (push) ve delete (pop) işlemleri, listenin sadece “top” adı verilen bir ucunda yani stack’in en üstünden gerçekleştirilir. Stack’e ait genel özellikler aşağıda sıralanmıştır;

- Yığındaki elemanlardan sadece en son eklenene erişim yapılır.
- Yığına ilk eklenen eleman en son elde edilir.
- FILO (First-in-Last-out) veya LIFO (Last-in-First-out) mantığıyla çalışır.
- İki tane temel işlem yapılabilir ;
  - ✓ push, yığının sonuna yeni bir eleman ekleme
  - ✓ pop, yığının en üstündeki elemanın alınması.



### Stack Veri Yapısının Örnek Kullanım Yerleri

Stack çok basit bir veri yapısı olmasına karşın birçok kullanım alanı bulunmaktadır. Aşağıda bazı kullanım alanlarına örnekler verilmiştir;

1. Word, Excel, Photoshop gibi yazılımlarda yapılan işlemlerin sırayla kayıt edildiği ve geri alınabilecek şekilde tutulduğu undo fonksiyonunda stack yapısı kullanılır.
2. Bir web tarayıcısında da ileri-geri adres gezmek için stack yapısı kullanılır.
3. Polish Notasyon: Infix olarak bilinen  $A*(B+C/D)-E$  cebirsel gösteriminin yerine hesap makinelerinde kullanılan postfix  $ABCD/+*E$  notasyonuna çevirme işleminde stack kullanılır.
4. Java derleyicisi program kodunun tamamını postfix'e çevirirken stack kullanılır.
5. Matematiksel bir ifadede veya program kodundaki parantez eşleştirme kontrolünü yapmak için stack kullanılır.
6. Java Virtual Machine (JVM) byte code'ları execute ederken altyapısında yine stack kullanılır.
7. Recursion ve function call işlemlerinin Bellekte gerçekleştirilmesinde stack kullanılır.
8. HTML/XML'de tag'lerin eşleştirilmesi bir stack uygulamasıdır.
9. Stack'ların bir diğer uygulama alanı labirent türü problemlerin çözümünde backtracking (bir yola gir yol tıkanırsa en son ki yol ayrımına geri gel başka yola devam et!) yöntemiyle kullanılır. Yol bilgisi bir stack yapısına push edilir yol yanlışsa son gidilen yanlış nokta pop edilir önceki noktaya devam edilir.

## Infix, Postfix ve Prefix Genel Gösterimi

Bilgisayarlarda infix yazım türünün çözülmesi zordur. Bu yüzden ifadelerin operatör önceliklerine göre ayrıştırılması, ayrılan parçaların sıralanması ve bu sıralamaya uyularak işlem yapılması gerekir. Bu işlemler için prefix ya da postfix notasyonu kullanılır. Çoğu derleyici, kaynak kod içerisinde infix notasyonunu kullanmaktadır ve daha sonra stack veri yapısını kullanarak prefix veya postfix notasyonuna çevirir. Aşağıdaki şekilde infix ifadelerin prefix ve postfix karşılıkları verilmiştir.

Infix	Postfix	Prefix
$A+B-C$	$AB+C-$	$-+ABC$
$(A+B)*(C-D)$	$AB+CD-*$	$*+AB-CD$
$A^B*C-D+E/F/(G+H)$	$AB^C*D-EF/GH+/+$	$+-*^ABCD//EF+GH$
$((A+B)*C-(D-E))^(F+G)$	$AB+C*DE-FG+^$	$^-*+ABC-DE+FG$
$A-B/(C*D^E)$	$ABCDE^*/-$	$-A/B*C^DE$

### Prefix : + a b

- ✓ İşlem işlenen işlenen (operator operand operand)
- ✓ İşlem sağdan sola doğru ilerler.
- ✓ Öncelik (parantez) yok.

### Infix : a + b

- ✓ İşlenen işlem işlenen (operand operator operand)
- ✓ İşlem öncelik sırasında göre ve soldan sağa doğru ilerler.

### Postfix : a b +

- ✓ İşlenen işlenen işlem (operand operand operator)
- ✓ İşlem soldan sağa doğru ilerler.
- ✓ Öncelik (parantez) yoktur.

Infix, prefix, postfix işlemlerinde dönüşüm için işlem önceliği dikkate alınır. İşlem öncelikleri şu şekildedir;

- Parantez
- Üs Alma
- Çarpma /Bölme
- Toplama/Çıkarma

Aynı işlem önceliğine sahip birden fazla ifadenin gelmesi durumunda, üs alma hariç, soldan sağa doğru işlemler yapılır. Üs almada sağdan sola doğrudur.  $A-B+C$  de öncelik  $(A-B)+C$  şeklindedir.  $A^B^C$  de ise  $A^(B^C)$  şeklindedir (Parantezler öncelik belirtmek için konulmuştur).

### Infix Gösterim

- **2+3\*5**
- + önce ise:
  - $(2+3)*5 = 5*5 = 25$
- \* önce ise:
  - $2+(3*5) = 2+15 = 17$

Infix gösterim paranteze ihtiyaç duyar. Burada operatörler operandların arasına gelir.

### Prefix Gösterim

- **A + B** → topla A B
- **+ 2 \* 3 5 =**
  - $= + 2 * 3 5 =$
  - $= + 2 15 = 17$
- **\* + 2 3 5 =**
  - $= * + 2 3 5 =$
  - $= * 5 5 = 25$

Prefixde iki operand arasındaki operatör önce belirtilir. Yani A ve B'yi toplamak için topla A B, çarpmak için ise çarp A B ile ifade edilir. Burada işlem sağdan sola doğru ilerler. Prefix'de paranteze ihtiyaç yoktur yani parantez kullanılmadan da işlem önceliği tanımlanabilir.

### Postfix Gösterim

- **A + B** → A B topla
- **2 3 5 \* + =**
  - $= 2 3 5 * + =$
  - $= 2 15 + = 17$
- **2 3 + 5 \* =**
  - $= 2 3 + 5 * =$
  - $= 5 5 * = 25$

Postfixde iki operand arasındaki operatör sonra belirtilir. Yani A ve B yi toplamak için A B topla, çarpmak için ise A B çarp ile ifade edilir. Burada işlem soldan sağa doğru ilerler. Postfix'de de paranteze ihtiyaç yoktur. Sonuç olarak Infix işlem sıralarının düzenlenmesi için paranteze ihtiyaç duyan tek gösterim şeklidir. Postfix forma çevrilen bir ifadede operand'ların bağlı olduğu operatör'leri (+, -, \*, /) görmek zor gelebilir. Burada birinci ifadenin sonucunun 17'e, ikinci ifadenin sonucunun ise 25'e karşılık geldiğini bulmak Infix gösterime alışık olduğumuz için zor gibi görünür. Fakat parantez kullanmadan tek anlama gelen hale dönüşür. İşlemleri, hesaplamaları yapmak kolaylaşır. Birçok derleyici bir Infix ifadenin değerini hesaplayacağı zaman Postfix forma dönüştürdükten yani belirsizliği ortadan kaldırdıktan sonra sonucu hesaplar.