



Bölüm 3. Çıkın

Olcay Taner Yıldız

2014

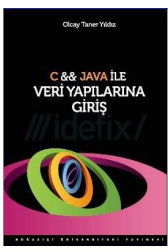


Sabit Dizi ile Çıkın Tanımı

Bağlı Liste ile Çıkın Tanımı

Uygulama: İfadeleri Hesaplama

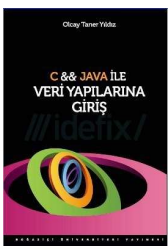
Sabit Dizi ile Çıkın Tanımı



Sabit dizi ile tanımlı bir çıkının içerdiği elemanların tanımı

Sabit Dizi ile Çıkın Tanımı	1
Bağlı Liste ile Çıkın Tanımı	2
Uygulama: İfadeleri Hesaplama	3
	4
	5
	6

```
public class Ornek{
    int icerik;
    public Ornek(int icerik){
        this.icerik = icerik ;
    }
}
```



Tam sayılar içeren bir çıkının sabit dizi ile tanımı

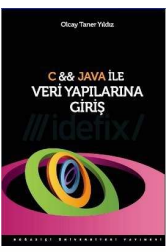
Sabit Dizi ile Çıkın Tanımı

Bağlı Liste ile Çıkın Tanımı

Uygulama: İfadeleri Hesaplama

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

```
public class Cikin{
    Ornek dizi [];
    int ust;
    int N;
    public Cikin(int N){
        dizi = new Ornek[N];
        this.N = N;
        ust = -1;
    }
    Ornek ust(){
        return dizi[ust];
    }
    boolean cikinDolu(){
        if (ust == N - 1)
            return true;
        else
            return false;
    }
    boolean cikinBos(){
        if (ust == - 1)
            return true;
        else
            return false;
    }
}
```

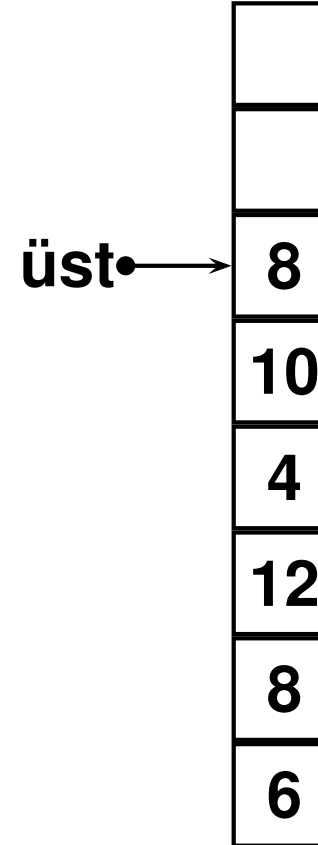


Altı elemandan oluşan büyüklüğü 8 olan bir çıkın yapısının dizi ile gösterimi

Sabit Dizi ile Çıkın Tanımı

Bağlı Liste ile Çıkın Tanımı

Uygulama: İfadeleri Hesaplama



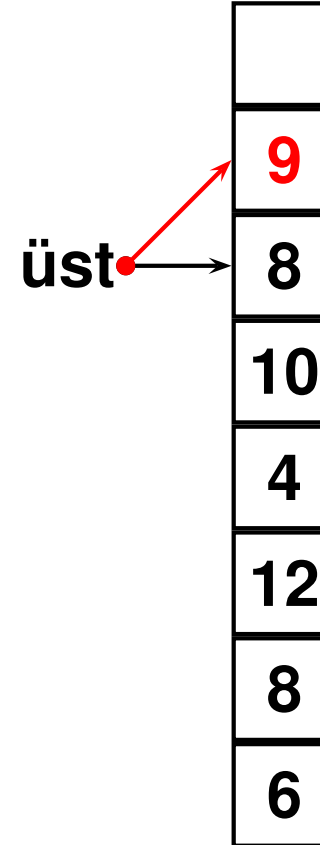


Çıkına eleman ekleme

Sabit Dizi ile Çıkın Tanımı

Bağlı Liste ile Çıkın Tanımı

Uygulama: İfadeleri Hesaplama





Çıkına yeni bir eleman ekleyen algoritma

Sabit Dizi ile Çıkın Tanımı	1
Bağlı Liste ile Çıkın Tanımı	2
Uygulama: İfadeleri Hesaplama	3
	4
	5
	6

```
void cikinEkle(Ornek yeni){  
    if (!cikinDolu ())  
        ust++;  
        dizi [ust] = yeni;  
    }  
}
```

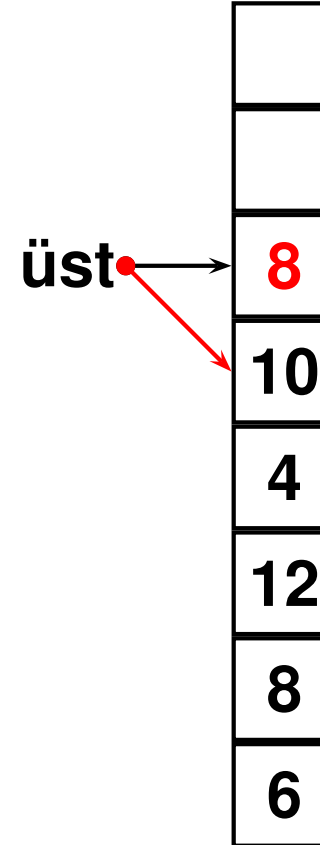


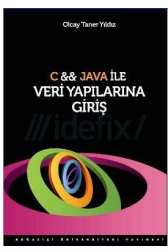
Çıkından eleman silme

Sabit Dizi ile Çıkın Tanımı

Bağlı Liste ile Çıkın Tanımı

Uygulama: İfadeleri Hesaplama





Çıkından bir eleman silen ve o elemanı döndüren algoritma

Sabit Dizi ile Çıkın Tanımı	1
Bağlı Liste ile Çıkın Tanımı	2
Uygulama: İfadeleri Hesaplama	3
	4
	5
	6
	7

```
Ornek cikinSil (){  
    if (!cikinBos ())  
        ust --;  
    return dizi[ust+1];  
}  
return null;  
}
```



Çıkm İşlemleri (Dizi)

Sabit Dizi ile Çıkm Tanımı

Baęlı Liste ile Çıkm Tanımı

Uygulama: İfadeleri Hesaplama

- Ekleme: $O(1)$
- Silme: $O(1)$

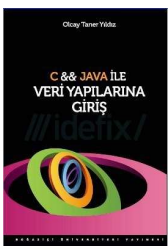


Sabit Dizi ile Çıkın Tanımı

Bağlı Liste ile Çıkın Tanımı

Uygulama: İfadeleri Hesaplama

Bağlı Liste ile Çıkın Tanımı



Tam sayılar içeren bir çıkının bağlı liste ile tanımı

Sabit Dizi ile Çıkın Tanımı

1

Bağlı Liste ile Çıkın Tanımı

2

3

Uygulama: İfadeleri Hesaplama

4

5

6

7

8

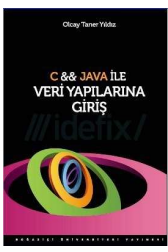
9

10

11

12

```
public class Cikin{
    Eleman ust;
    public Cikin(){
        ust = null;
    }
    boolean cikinBos(){
        if (ust == null)
            return true;
        else
            return false;
    }
}
```

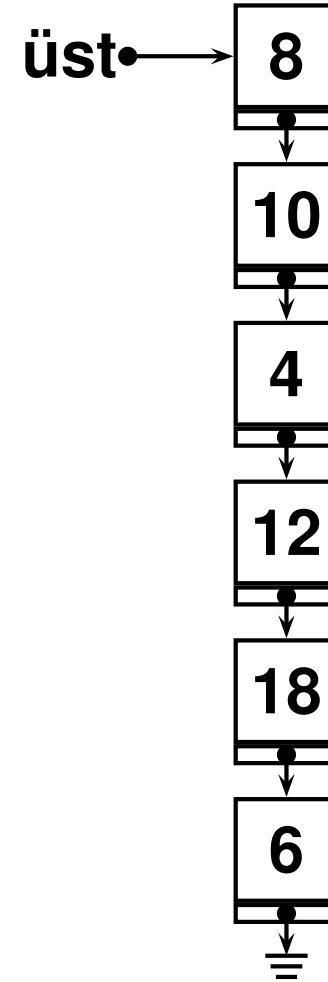


Altı elemandan oluşan bir çıkın yapısının bağlı liste ile gösterimi

Sabit Dizi ile Çıkın Tanımı

Bağlı Liste ile Çıkın Tanımı

Uygulama: İfadeleri Hesaplama



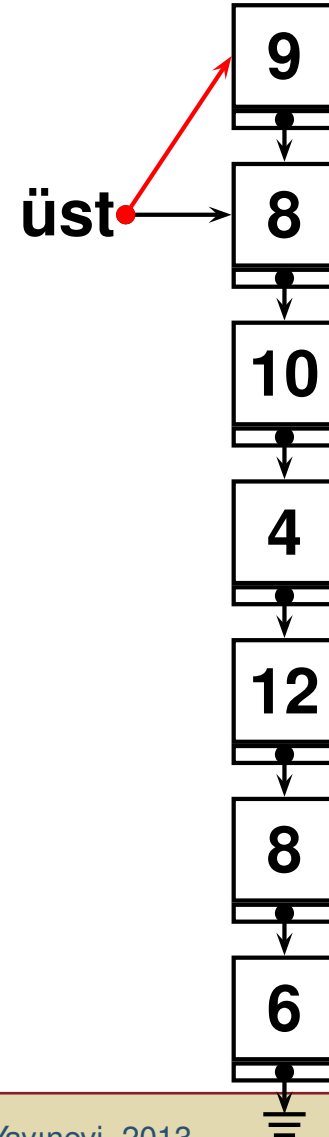


Çıkına eleman ekleme

Sabit Dizi ile Çıkın Tanımı

Bağlı Liste ile Çıkın Tanımı

Uygulama: İfadeleri Hesaplama





Çıkına yeni bir eleman ekleyen algoritma

Sabit Dizi ile Çıkın Tanımı	1
Bağlı Liste ile Çıkın Tanımı	2
Uygulama: İfadeleri Hesaplama	3
	4

```
void cikinEkle(Eleman yeni){  
    yeni. ileri = ust;  
    ust = yeni;  
}
```

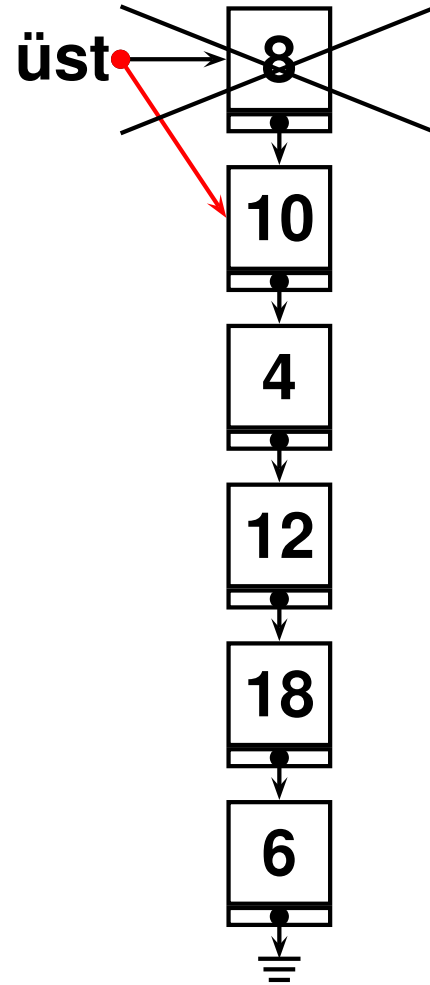


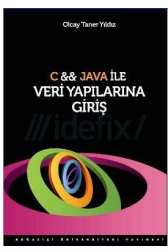
Çıkından eleman silme

Sabit Dizi ile Çıkın Tanımı

Bağlı Liste ile Çıkın Tanımı

Uygulama: İfadeleri Hesaplama





Çıkından bir eleman silen ve o elemanı döndüren algoritma

Sabit Dizi ile Çıkın Tanımı	1
Bağlı Liste ile Çıkın Tanımı	2
Uygulama: İfadeleri Hesaplama	3
	4
	5
	6

```
Eleman cikinSil (){\n    Eleman e = ust;\n    if (!cikinBos ())\n        ust = ust. ileri ;\n    return e;\n}
```



Çıkın İşlemleri (Bağlı Liste)

Sabit Dizi ile Çıkın Tanımı

Bağlı Liste ile Çıkın Tanımı

Uygulama: İfadeleri Hesaplama

- Ekleme: $O(1)$
- Silme: $O(1)$

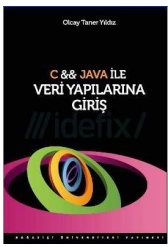


Sabit Dizi ile Çıkın
Tanımı

Bağlı Liste ile Çıkın
Tanımı

Uygulama: İfadeleri
Hesaplama

Uygulama: İfadeleri Hesaplama



Sabit Dizi ile Çıkmın Tanımı

Bağlı Liste ile Çıkmın Tanımı

Uygulama: İfadeleri Hesaplama

Matematiksel İfadeler

- Belirli bir matematiksel ifadeyi bir değişkene atayan kodlar

```
x = a / b - c + d * e - a * c;
```

- Belirli bir matematiksel ifadenin doğru olup olmadığını belirleyen kodlar

```
if (3 * x - 5 > 4 * y * y)
```



Matematiksel İfadeler

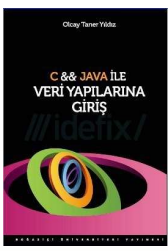
Sabit Dizi ile Çıkın Tanımı

Bağlı Liste ile Çıkın Tanımı

Uygulama: İfadeleri Hesaplama

$$a/b - c + d * e - a * c$$

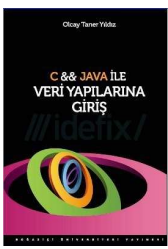
- İşlemler: /, -, +, *, -, *
- İşlenenler: a , b , c , d ve e



Matematiksel ifadenin elemanlarını içeren veri yapısı (1)

Sabit Dizi ile Çıkmın Tanımı	1
Bağlı Liste ile Çıkmın Tanımı	2
Uygulama: İfadeleri Hesaplama	3
	4
	5
	6
	7
	8
	9

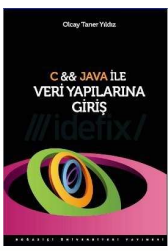
```
public class Ornek{
    int tip;
    int islenen;
    char islem;
    int oncelik;
    public Ornek(int islenen){
        this.tip = 0;
        this.islenen = islenen;
    }
}
```



Matematiksel ifadenin elemanlarını içeren veri yapısı (2)

Sabit Dizi ile Çıkmın Tanımı	10
Bağlı Liste ile Çıkmın Tanımı	11
Uygulama: İfadeleri Hesaplama	12
	13
	14
	15
	16
	17
	18
	19
	20
	21
	22
	23
	24
	25
	26

```
public Ornek(char islem){
    this.tip = 1;
    this.islem = islem;
    switch (islem){
        case '(':oncelik = 0;
            break;
        case '+':
        case '-':oncelik = 1;
            break;
        case '*':
        case '/':oncelik = 2;
            break;
        case ')':oncelik = 3;
            break;
    }
}
```



Arka Gösterim

Sabit Dizi ile Çıkmın Tanımı

Bağlı Liste ile Çıkmın Tanımı

Uygulama: İfadeleri Hesaplama

$ab/c - de * +ac * -$ arka gösteriminin değerinin hesaplanması aşağıda gösterilmiştir.

$T_1 = a / b$	$T_1 \ c - d e * + a c * -$
$T_2 = T_1 - c$	$T_2 \ d e * + a c * -$
$T_3 = d * e$	$T_2 \ T_3 + a c * -$
$T_4 = T_2 + T_3$	$T_4 \ a c * -$
$T_5 = a * c$	$T_4 \ T_5 -$
$T_6 = T_4 - T_5$	T_6



Arka gösterimi verilen bir ifadenin değerini hesaplayan algoritma (1)

Sabit Dizi ile Çıkmın Tanımı

Bağlı Liste ile Çıkmın Tanımı

Uygulama: İfadeleri Hesaplama

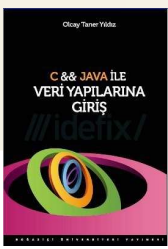
```
1  int hesapla(Ornek[] ifade){
2      int i;
3      Ornek e, e1, e2, s;
4      Cikin c = new Cikin(100);
5      for (i = 0; i < ifade.length; i++){
6          e = ifade[i];
7          if (e.tip == 0){
8              c.cikinEkle(e);
9          }else{
10             e2 = c.cikinSil ();
11             e1 = c.cikinSil ();
12             s = islem(e.islem, e1.islenen, e2.islenen);
13             c.cikinEkle(s);
14         }
15     }
16     e = c.cikinSil ();
17     return e.islenen;
18 }
```



Arka gösterimi verilen bir ifadenin değerini hesaplayan algoritma (2)

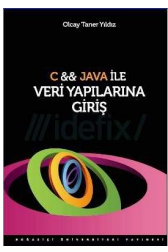
Sabit Dizi ile Çıkmın Tanımı	19
Bağlı Liste ile Çıkmın Tanımı	20
Uygulama: İfadeleri Hesaplama	22
	23
	24
	25
	26
	27
	28
	29
	30
	31
	32

```
Ornek islem(char ch, int e1, int e2){
    int sonuc;
    switch (ch){
        case '+':sonuc = e1 + e2;
            break;
        case '-':sonuc = e1 - e2;
            break;
        case '*':sonuc = e1 * e2;
            break;
        case '/':sonuc = e1 / e2;
            break;
    }
    return new Ornek(sonuc);
}
```



Arka gösterimi $ab/c - de * + a*$ olan bir ifadenin çıkın yardımıyla hesaplanması

\boxed{a} $b / c - de^* + a^*$	$\begin{array}{c} \boxed{b} \\ \boxed{a} \end{array}$ $/ c - de^* + a^*$	$T_1 = a/b$ $\boxed{T_1}$ $c - de^* + a^*$	$\begin{array}{c} \boxed{c} \\ \boxed{T_1} \end{array}$ $- de^* + a^*$
$T_2 = T_1 - c$ $\boxed{T_2}$ $de^* + a^*$	$\begin{array}{c} \boxed{d} \\ \boxed{T_2} \end{array}$ $e^* + a^*$	$\begin{array}{c} \boxed{e} \\ \boxed{d} \\ \boxed{T_2} \end{array}$ $* + a^*$	$T_3 = d * e$ $\begin{array}{c} \boxed{T_3} \\ \boxed{T_2} \end{array}$ $+ a^*$
$T_4 = T_2 + T_3$ $\boxed{T_4}$ a^*	$\begin{array}{c} \boxed{a} \\ \boxed{T_4} \end{array}$ $*$	$T_5 = a * T_4$ $\boxed{T_5}$	



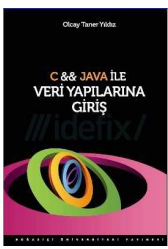
Arka gösterimi $ab/c - de * +a*$ olan bir ifadenin çıkın yardımıyla hesaplanması

Sabit Dizi ile Çıkın Tanımı

Bağlı Liste ile Çıkın Tanımı

Uygulama: İfadeleri Hesaplama

1. a bir işlenendir. Çıkına eklenir.
2. b bir işlenendir. Çıkına eklenir.
3. $/$ bir işlemdir. Çıkından iki eleman silinir (a ve b), bu iki elemanla $/$ işlemi yapıp sonuç (T_1) çıkına eklenir.
4. c bir işlenendir. Çıkına eklenir.
5. $-$ bir işlemdir. Çıkından iki eleman silinir (T_1 ve c), bu iki elemanla $-$ işlemi yapıp sonuç (T_2) çıkına eklenir.
6. d bir işlenendir. Çıkına eklenir.
7. e bir işlenendir. Çıkına eklenir.
8. $*$ bir işlemdir. Çıkından iki eleman silinir (d ve e), bu iki elemanla $*$ işlemi yapıp sonuç (T_3) çıkına eklenir.
9. $+$ bir işlemdir. Çıkından iki eleman silinir (T_2 ve T_3), bu iki elemanla $+$ işlemi yapıp sonuç (T_4) çıkına eklenir.
10. a bir işlenendir. Çıkına eklenir.
11. $*$ bir işlemdir. Çıkından iki eleman silinir (a ve T_4), bu iki elemanla $*$ işlemi yapıp sonuç (T_5) çıkına eklenir.



Verilen bir ifadenin arka gösterimini bulan algoritma (1)

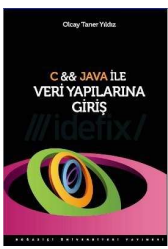
Sabit Dizi ile Çıkmın Tanımı

Bağlı Liste ile Çıkmın Tanımı

Uygulama: İfadeleri Hesaplama

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

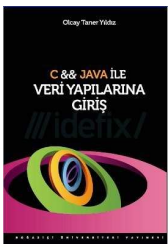
```
void arkaGosterim(Ornek[] ifade){
    int i;
    Ornek e, e1;
    Cikin c = new Cikin(100);
    for (i = 0; i < ifade.length; i++){
        e = ifade[i];
        if (e.tip == 0)
            System.out.print(e.islenen);
        else
            if (e.islem == '(')
                c.cikinEkle(e);
            else
                if (e.islem == ')'){
                    e1 = c.cikinSil ();
                    while (e1.islem != '('){
                        System.out.print(e1.islem);
                        e1 = c.cikinSil ();
                    }
                }
    }
}
```



Verilen bir ifadenin arka gösterimini bulan algoritma (2)

Sabit Dizi ile Çıkın Tanımı	20
Bağlı Liste ile Çıkın Tanımı	21
Uygulama: İfadeleri Hesaplama	23
	24
	25
	26
	27
	28
	29
	30
	31
	32

```
        else{
            while (e.oncelik <= c.ust().oncelik){
                e1 = c.cikinSil ();
                System.out.print(e1.islem);
            }
            c.cikinEkle (e);
        }
    }
    while (!c.cikinBos()){
        e1 = c.cikinSil ();
        System.out.print(e1.islem);
    }
}
```



Ara gösterimini $A + B * (C + D)$ olan bir ifadenin arka gösteriminin bir çıkın yardımıyla bulunması

Sabit Dizi ile Çıkın Tanımı

Bağlı Liste ile Çıkın Tanımı

Uygulama: İfadeleri Hesaplama

	+	+	*	(
A	+	B	*	(
A	A	AB	AB	AB
(+	+	*	
*	((*	
+	*	*	+	
C	+	D)	
ABC	ABC	ABCD	ABCD+	ABCD+*+



Sabit Dizi ile Çıkın Tanımı

Bağlı Liste ile Çıkın Tanımı

Uygulama: İfadeleri Hesaplama

*Ara gösterimini $A + B * (C + D)$ olan bir ifadenin arka gösteriminin bir çıkın yardımıyla bulunması*

1. A bir işlenendir. Ekranaya yazılır.
2. + bir işlemdir. Çıkında eleman olmadığından çıkına eklenir.
3. B bir işlenendir. Ekranaya yazılır.
4. * bir işlemdir. *'in önceliği çıkının en üstündeki +'dan büyük olduğundan çıkına eklenir.
5. (bir işlemdir. (doğrudan çıkına eklenir.
6. C bir işlenendir. Ekranaya yazılır.
7. + bir işlemdir. +'nın önceliği çıkının en üstündeki ('den büyük olduğundan çıkına eklenir.
8. D bir işlenendir. Ekranaya yazılır.
9.) bir işlemdir. Çıkından (görene kadar işlemler silinir ve ekranaya yazılır.
10. Çıkında kalan * ve + sırasıyla çıkından silinip ekranaya yazılır.