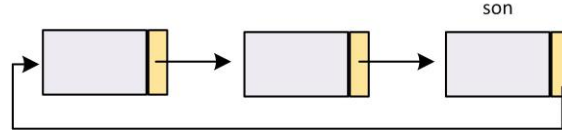


TEK YÖNLÜ DAİRESEL BAĞLI LİSTELER

Tek yönlü dairesel bağlı listelerin, tek yönlü bağlı listelerden temel farkı son elemanın sonraki değerinin ilk elemanı göstermesidir. Şekil 1’de tek yönlü dairesel bağlı liste örneği verilmiştir.



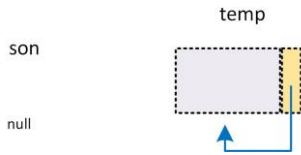
Şekil 1. Tek Yönlü Dairesel Bağlı Liste

Şekilden de görüleceği üzere tek yönlü bağlı liste düğüm yapısı kullanılmaktadır. Bu düğüm yapısı Java’da aşağıdaki şekilde tanımlanabilir.

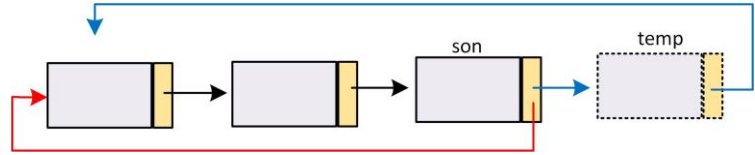
```
public class Dugum {  
    Dugum next;  
    int deger;  
}
```

Tek yönlü dairesel bağlı liste için son elemanın tutulması yeterli olmaktadır. Son.next dediğimiz zaman ilk elemanı elde etmek mümkündür. Ekleme işlemi şekil 2’deki gibi şekilde gerçekleştirilir. Burada listenin boş olma durumu için farklı işlem yapmak gerekir. Şekil 2’nin hemen altında bu listeye eleman eklemek için gerekli kod gösterilmiştir.

Boş Listeye Ekleme



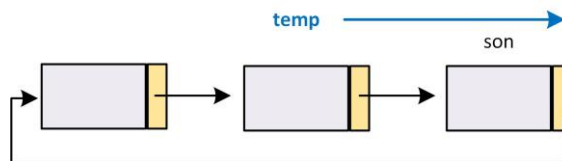
N. ELEMANI EKLEME



Şekil 2. Tek Yönlü Dairesel Bağlı Liste Eleman Ekleme

```
void ekle(int eleman) {  
    Dugum temp = new Dugum();  
    temp.deger = eleman;  
    if(son == null) {  
        temp.next = temp;  
        son = temp;  
    }else {  
        temp.next = son.next;  
        son.next = temp;  
        son = temp;  
    }  
}
```

Elemanlarda gezinmek için ilk elemanın sonrasından başlanır. İlk elemana ayrıca işlem uygulanır. Geçici değişkenimiz ilk eleman olduğu zaman döngü sonlandırılır. Tek yönlü dairesel bağlı listelerde gezinme işlemi şekil 3 ile verilmiştir.



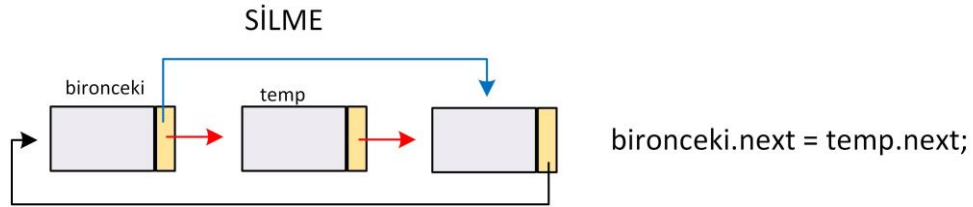
Şekil 3. Tek Yönlü Dairesel Bağlı Listede Gezinme

Şekil 3'teki mantıkla listeleme ve arama işlemleri gerçekleştirilir. Listeleme ve arama kodları aşağıda verilmiştir.

```
void listele() {
    Dugum temp = son.next.next;
    System.out.println(son.next.deger);
    while(temp != son.next) {
        System.out.println(temp.deger);
        temp = temp.next;
    }
}

Dugum ara(int eleman) {
    Dugum temp = son.next.next;
    if(son.next.deger == eleman) {
        return son.next;
    }else {
        while(temp!=son.next) {
            if(temp.deger == eleman) {
                return temp;
            }
        }
    }
    return null;
}
```

Silme işlemi için bir önceki düğümün nexti silinecek elemanın nextine eşitlenmelidir. Bu sayede şekil 4'teki gibi kırmızı ile gösterilen bağlar koparılacak mavi ile gösterilen bağlar eklenmiş olur. Silme işleminin kodu aşağıda verilmiştir.



Şekil 4. Tek Yönlü Dairesel Bağlı Listede Silme İşlemi

```
void sil(int eleman) {
    Dugum temp = son.next.next;
    Dugum bironceki = son.next;
    if(son.next.deger == eleman) {
        son.next = son.next.next;
    }else {
        while(temp != son.next) {
            if(temp.deger == eleman) {
                bironceki.next = temp.next;
                if(temp == son) {
                    son = bironceki;
                }
            }
            bironceki = temp;
            temp = temp.next;
        }
    }
}
```