

İzmir Bakırçay Üniversitesi
Mühendislik ve Mimarlık Fakültesi
Bilgisayar Mühendisliği Bölümü
Veri Yapıları Dersi

Vize Ödevi: Döngüsel Linked list ile Patlamayan Balonu Bulma

Caner Çakmak

200601070

```
package com.mycompany._caner_cakmak;  
import java.util.Scanner;
```

Öncelikle balon sayısını ve patlatılacak balon pozisyonunu belirten değerleri kullanıcıdan alabilmek için java için kullanılan Scanner kütüphanesini projemize ekliyoruz.

```
public class App {  
    public static class Node {  
        int data;  
        Node next;  
  
        public Node(int data) {  
            this.data = data;  
            this.next = null;  
        }  
    }  
}
```

Ardından projemizdeki Linked List için kullanacağımız Node class'ını da oluşturuyoruz. Bu class içerisinde her node 'un kendine özel datası ve sonraki Node' un adresini tutmak ve daha sonra buna ulaşmak için kullanacağı next pointer'ı bulmakta. Ekstra olarak bu class içinde bir constructor metodu kullanarak Node tanımlanırken data atamasını ve nextinin null olduğunu burada tanımlıyoruz.

```

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Balon sayisini giriniz: ");
    int n = scanner.nextInt();
    System.out.print("Atlanacak balon sayisini giriniz: ");
    int m = scanner.nextInt();

    balonPozisyonuBul(m, n);
}

```

Projemizin Main fonksiyonunda projeye dahil ettiğimiz Scanner kütüphanesi yardımıyla kullanıcıdan aldığımız balon sayısını “n” değişkenine ve patlatılmak istenen balonun pozisyonunu ise “m” değişkenine atarak kullanıcıdan gerekli inputları almış oluyoruz.

Devamında ise projemizdeki ana fonksiyon olan balonPozisyonuBul fonksiyonuna gerekli “n” ve “m” değerlerini göndererek bu fonksiyonun çıktısını bekliyoruz.

```

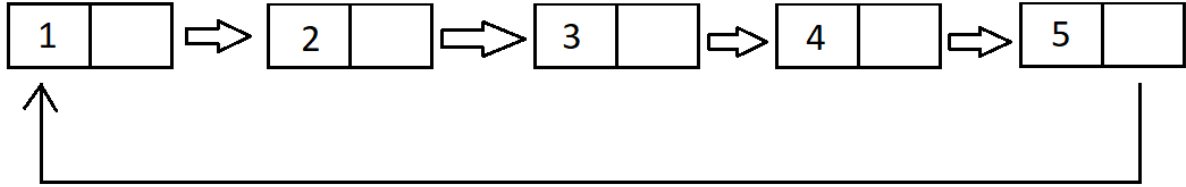
static void balonPozisyonuBul(int m, int n) {
    // N adet balondan oluşacak dairesel bağlı listeyi oluşturma
    Node head = new Node(data: 1);
    Node current = head;
    for (int i = 2; i <= n; i++)
    {
        current.next = new Node(data: i);
        current = current.next;
    }

    current.next = head; // son düğümün gösterdiği next düğüm head'e bağlanır
}

```

Projemizin ana fonksiyonu olan balonPozisyonuBul adlı bu fonksiyonda projeye dahil ettiğimiz Scanner kütüphanesiyle aldığımız inputların bu fonksiyona gönderilmesiyle beraber öncelikle bir “head” yani başlangıç düğümü oluşturuyoruz. Sonra bu düğümü şimdiki düğüm olarak yani “current” düğüm olarak belirliyoruz. 1. Düğümümüz head olduğundan dolayı for döngüsü 2 den başlayarak yeni düğümler oluşturuyor her bir düğümü birbirine bağlıyor ve bu

işlemi girilen n sayısı kadar yapıyor. En sonda ise son düğümün nextini head düğümüne bağlıyoruz ve bu sayede döngüsel bir düğüm oluşturmuş oluyoruz.



Bu şekilde bütün düğümler birbirini göstermekte ve bu sayede gezinti yapabilmekteyiz.

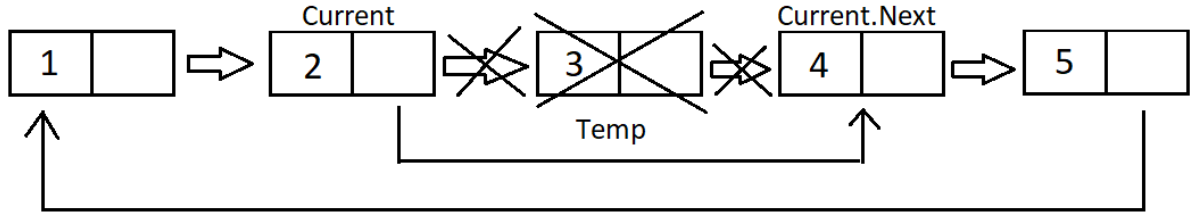
```
// Listenin sonunda kalacak balonu bulma
while (current.next != current) {
    // M-1 balon atlanır
    for (int i = 0; i < m - 1; i++) {
        current = current.next;
    }
    // m-1 den sonra gelen balon patlatılır
    Node temp = current.next;
    current.next = temp.next;
    temp = null;
}

System.out.println("Patlamayan Balonun Pozisyonu: " + current.data);
```

Şimdiki düğümümüzün gösterdiği next düğüm eğer şimdiki düğümümüze eşitse yani elimizde tek bir düğüm kaldıysa döngüden çıkacak bir döngü kuruyoruz.

Bu döngü0 dan başlayarak patlatılacak balonun pozisyonundan bir önceki pozisyonda duruyor ve bu pozisyonu şimdiki yani “current” olarak adlandırıyor.

Sonra bu şimdiki düğümünden sonra gelen düğümü temp düğümüne atıyor. Temp düğümünden bir sonraki düğümü ise şimdiki düğümün next’ine atıyor ve temp düğümünü null olarak işaretliyor. Böylelikle kaldırılmak istenen pozisyondaki düğüm ortadan kalkmış oluyor ve aradaki bağlantı sanki o düğüm orda hiç olmamış gibi tamamlanıyor.



Bu şekilde elimizde tek bir düğüm kalana kadar devam ediyor ve en son düğümün datasını yazdırarak programımız sona eriyor.

```
Balon sayisini giriniz: 5
Atlanacak balon sayisini giriniz: 3
Patlamayan Balonun Pozisyonu: 4
```

$N = 5$ ve $M = 3$ için çıktımız 4 olmaktadır.