

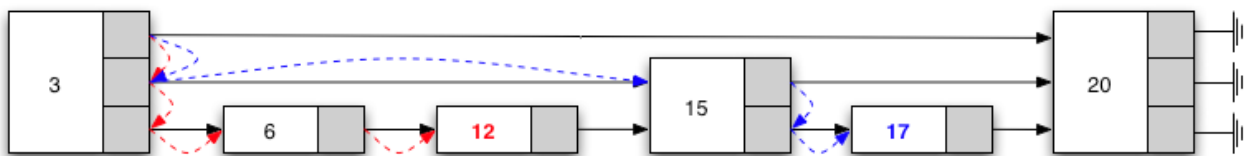
Rekursive Strukturen

Datenstrukturen und Algorithmen

Inhaltsverzeichnis

Skiplist.....	1
Eigene Struktur anlegen.....	2
Josephus.....	3

Skiplist



Rot dargestellt:

```
node = new Node();
node.value = 12;
node.next = new Node[1];
head.next[0].next[0] = node;
node.next[0] = head.next[1];
```

Blau dargestellt:

```
node = new Node();
node.value = 17;
node.next = new Node[1];
head.next[1].next[0] = node;
node.next[0] = head.next[2];
```

Ausgabe: 3 15 20

```
node = head;
while (node != null) {
    System.out.println(node.value);
    node = node.next[1];
}
```

Eigene Struktur anlegen

```

public class Continent {

    public String name;
    public double area;
    public long population;

}

public class ContinentTest {

    public static double populationDensity(Continent c) {
        return c.population/c.area;
    }

    public static void print(Continent c) {
        System.out.println("Name: " + c.name);
        System.out.println("Bevölkerung: " + c.population);
        System.out.println("Fläche: " + c.area);
        System.out.println("Bevölkerungsdichte: " + populationDensity(c));
        System.out.println();
    }

    public static void main(String[] args) {

        Continent[] continents = new Continent[5];

        continents[0] = new Continent();
        continents[0].name = "Europa";
        continents[0].area = 10.5;
        continents[0].population = 733;
        continents[1] = new Continent();
        continents[1].name = "Afrika";
        continents[1].area = 30.3;
        continents[1].population = 944;
        continents[2] = new Continent();
        continents[2].name = "Asien";
        continents[2].area = 44.4;
        continents[2].population = 4010;
        continents[3] = new Continent();
        continents[3].name = "Amerika";
        continents[3].area = 42.7;
        continents[3].population = 904;
        continents[4] = new Continent();
        continents[4].name = "Ozeanien";
        continents[4].area = 8.5;
        continents[4].population = 34;

        for (int i=0; i<continents.length; ++i) {
            print(continents[i]);
        }

    }

}

```

Josephus

```

public class Node {

    public int value;
    public Node next;

}

import java.util.Scanner;

public class Josephus {

    public static int josephus(int n, int s) {

        // n Knoten erzeugen
        Node head = new Node();
        head.value = 1;
        Node node = head;
        for (int i=2; i<=n; ++i) {
            node.next = new Node();
            node = node.next;
            node.value = i;
        }
        // letzter Knoten verweist wieder auf den ersten
        node.next = head;

        // Knoten entfernen
        node = head;
        while (node != node.next) {
            for (int i=1; i<s; ++i) {
                node = node.next;
            }
            node.next = node.next.next;
        }

        return node.value;

    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Anzahl Objekte: ");
        int n = scanner.nextInt();
        System.out.print("Schrittweite: ");
        int s = scanner.nextInt();

        System.out.println("Letztes Objekt: "+josephus(n,s));

        scanner.close();
    }
}

```