Rekursive Strukturen

Datenstrukturen und Algorithmen

Inhaltsverzeichnis

Skiplist

Das folgende Java-Programm verwaltet eine sogenannte Skip-Liste. Bei dieser besteht die Möglichkeit, dass Knoten neben dem Verweis auf den nachfolgenden Knoten zusätzliche Verweise auf weiter entfernte Knoten haben und somit andere Knoten überspringen.

Was gibt das Programm aus und warum? Zeichnen Sie die Situation im Speicher.

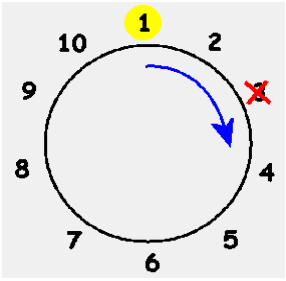
```
public class Node {
	int value;
	Node[] next;
}


public class Skiplist {

	public static void main(String[] args) {

		Node head;
		Node node;

		node = new Node();
		node.value = 3;
		node.next = new Node[3];
		head = node;

		node = new Node();
		node.value = 6;
		node.next = new Node[1];
		head.next[0] = node;

		node = new Node();
		node.value = 15;
		node.next = new Node[2];
		head.next[1] = node;
		head.next[0].next[0] = node;
```

```
		node = new Node();
		node.value = 20;
		node.next = new Node[3];
		head.next[2] = node;
		head.next[1].next[1] = node;
		head.next[1].next[0] = node;

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

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


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

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

	}

}
```

Eigene Struktur anlegen

Legen Sie eine neue Struktur (in Java/Eclipse entspricht das einer Klasse) `Continent` an. Es sollte je ein Attribut für Name, Fläche und Einwohnerzahl vorhanden sein. Schreiben Sie ein Java-Programm mit main-Methode, in dem Sie für jeden Kontinent eine Instanz der Klasse `Continent` anlegen (als Array, also eine Variable vom Datentyp `Continent[]`). Das Programm soll alle Informationen zu den Kontinenten auf der Konsole ausgeben. Schreiben Sie dazu eine Methode `public static void print(Continent c)`, die alle Informationen zu einem Kontinent auf der Konsole ausgibt. Schreiben Sie eine Methode `public static double populationDensity(Continent c)`, die die Bevölkerungsdichte von einem Kontinent berechnet und zurück gibt. Verwenden Sie diese in der main-Methode.

Josephus



Schreiben Sie ein Java-Programm, das das Josephus-Problem für zwei vom Benutzer einzugebende ganze Zahlen `n` und `s` löst. Bei diesem Problem werden `n` (nummerierte) Objekte in einem Kreis angeordnet. Dann wird jedes `s`-te Objekt aus dem Kreis entfernt, wobei der Kreis immer wieder geschlossen wird, bis nur noch ein Objekt übrig ist. Welches Objekt bleibt als letztes übrig (mit welcher Nummer)?

Im Bild oben rechts ist `n=10` und `s=2`.

Sie können das folgende Programm als Vorlage verwenden und die Methode schreiben.

```
import java.util.Scanner;

public class Josephus {

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

	}

	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();
	}

}
```

Tipp: Verwenden Sie eine einfach verkettete Liste, bei der der letzte Knoten wieder auf den ersten Knoten verweist. Entfernen Sie dann so lange Knoten aus diesem Kreis bis nur noch ein Knoten vorhanden ist. Das ist zwar die elegantere Variante, Sie dürfen es aber auch mit einem Array versuchen.