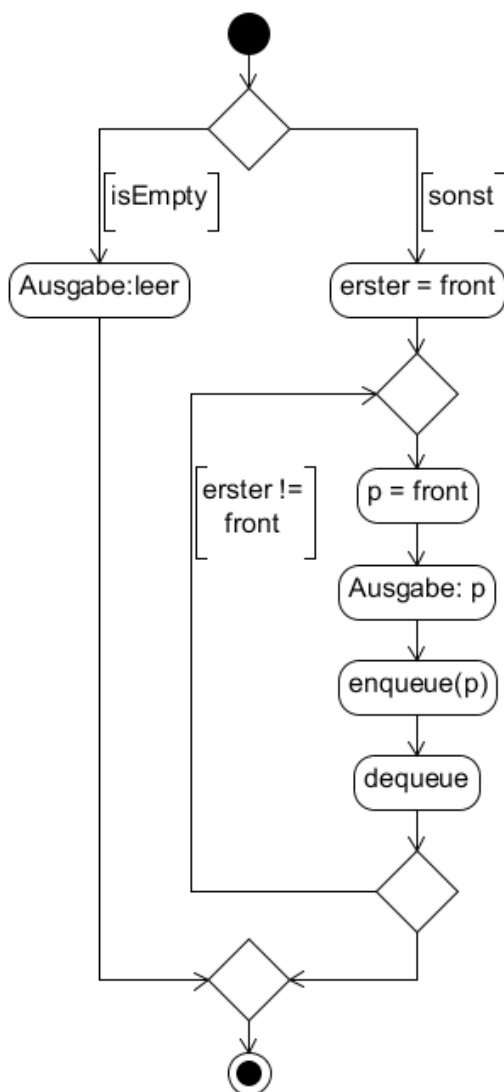


## Nassi-Shneidermann-Diagramme

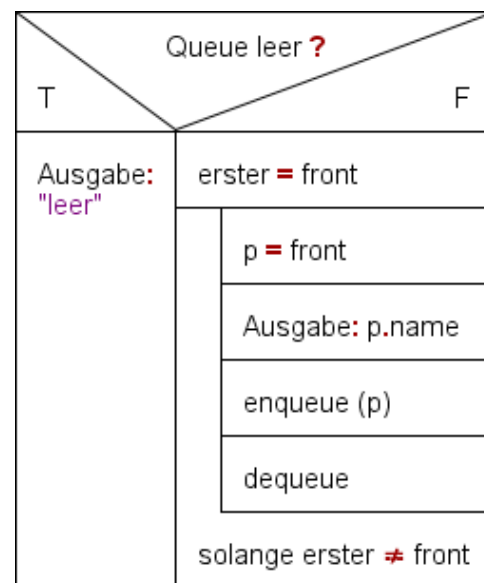
Aktivitätsdiagramme sind nicht die einzige Möglichkeit, Algorithmen darzustellen. Bereits 1972 wurde eine Diagrammtechnik von Isaac Nassi und Ben Shneidermann entwickelt und nach ihnen benannt<sup>1</sup>. Auf deutsch werden diese Diagramme auch „Struktogramme“ genannt. Sie wirken mittlerweile etwas altbacken und gehören nicht zum UML-Standard, werden aber für das Zentralabitur in NRW noch gefordert.

Als Beispiel betrachten wir den Algorithmus zur Ausgabe aller Elemente einer Queue:

### Aktivitätsdiagramm:



### Struktogramm:



### Quellcode:

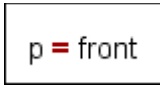
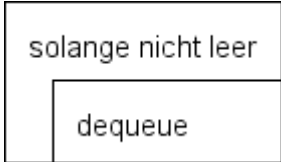
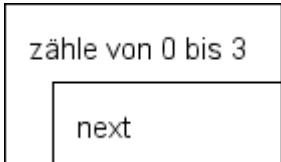
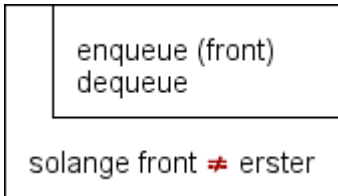
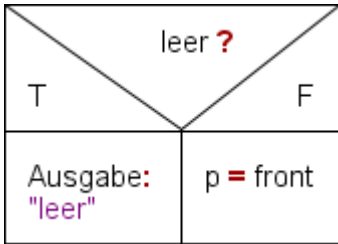
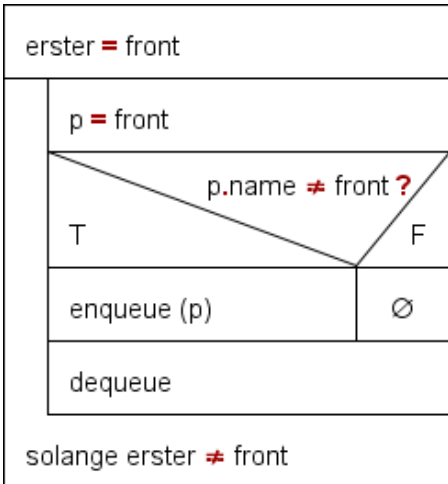
```

public void ausgeben()
{
    Patient p, erster;
    if (queue.isEmpty())
    {
        Console.println("leer");
    }
    else
    {
        erster = queue.front();
        do {
            p = queue.front();
            Console.println(p.getName());
            queue.enqueue(p);
            queue.dequeue();
        } while (queue.front()
                != erster);
    }
}
  
```

<sup>1</sup> Siehe auch <http://de.wikipedia.org/wiki/Nassi-Shneiderman-Diagramm>



## Elemente von Struktogrammen

<b>Einfache Anweisung</b> <code>p = queue.front();</code>	
<b>While-Schleife</b> <code>while (!queue.isEmpty())  {      queue.dequeue();  }</code>	
<b>For-Schleife</b> <code>for (i = 0; i &lt; 4; i++)  {      liste.next();  }</code>	
<b>Do-While-Schleife</b> <code>do  {      queue.enqueue(queue.front());      queue.dequeue();  } while (queue.front() != erster)</code>	
<b>Verzweigung</b> <code>if (queue.isEmpty())  {      Console.println("leer");  }  else  {      p = queue.front();  }</code>	
<b>Kombinationen</b>  Alle Struktogrammelemente haben außen die Form eines Rechtecks. Sie können so beliebig hintereinander gesetzt und ineinander geschachtelt werden.  In diesem Beispiel ist z.B. eine Verzweigung in eine do-while-Schleife geschachtelt.	

## Struktogramm-Editor

Ein brauchbarer Open-Source-Editor für Struktogramme ist der Structorizer:  
<https://structorizer.fisch.lu/>



## Aufgabe 1

Stelle den folgenden Quellcode als Struktogramm dar:

```
public void kontaktEinfügen(Kontakt pNeu)
{
    Kontakt kAkt;

    liste.toFirst();
    while (liste.hasAccess())
    {
        kAkt = liste.getContent();
        if (pNeu.getNachname().compareTo(kAkt.getNachname()) < 0)
        {
            liste.insert(pNeu);
            break;
        }
        liste.next();
    }
    if (liste.hasAccess() == false)
    {
        liste.append(pNeu);
    }
}
```

## Aufgabe 2

Beschreibe den Algorithmus, der im rechts abgebildeten Struktogramms dargestellt wird, mit eigenen Worten.

