

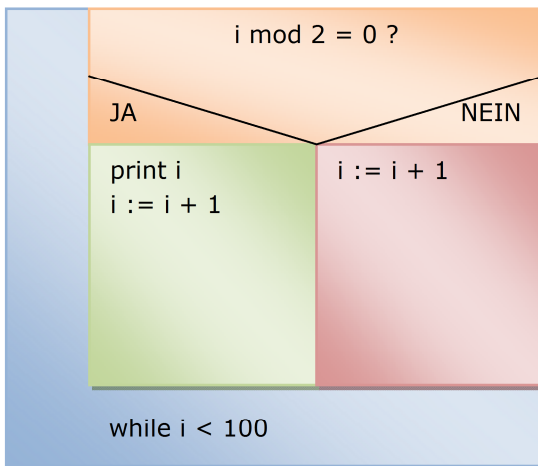
# ÜBERSICHTSBLATT: PROGRAMME PLANEN

## Warum das ganze?

Bevor man anfängt ein Programm zu schreiben, sollte man sich Gedanken darüber machen, wie das Programm ablaufen sollte. Auf diese Weise kann man logisch zusammenhängende Funktionsblöcke identifizieren und in wiederverwendbaren Modulen unterbringen.

Es gibt verschiedene Möglichkeiten Programme zu planen. Einiger der populärsten sollen hier kurz vorgestellt werden.

## Nassi-Schneidermann Diagramme



Die Nassi-Schneidermann Diagramme (oft auch einfach als Struktogramme bezeichnet) sind eine recht übersichtliche Möglichkeit die Funktion eines Programmes darzustellen.

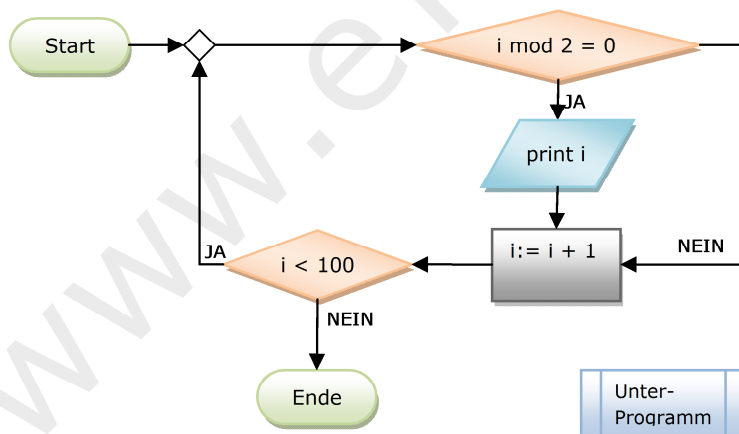
Es werden verschiedene Typen von Strukturelementen in eine Graphik eingebaut die dabei für verschiedene Programmbestandteile stehen.

Im nebenstehenden Beispiel wird ein Programm beschrieben, das alle geraden Zahlen ausgibt die kleiner sind als 100. Wir sehen eine fußgesteuerte Schleife, eine

Verzweigung und zwei Sequenzen.

Das Problem mit diesen Diagrammen besteht allerdings darin, dass sie bei größeren Programmen schnell in tapetenartige Formate abschweifen. Darüber hinaus lassen sie sich nicht leicht ändern und sind nur mit Toolunterstützung praktikabel einsetzbar.

## Programmablaufplan



Eine wesentlich intuitivere Variante der Programmplanung ist der sogenannte Programmablaufplan. Für verschiedene Anweisungstypen gibt es hier verschiedene Symbole die mit Pfeilen verbunden werden.

Diese Pläne lassen sich auch in Office-Anwendungen (z.B. Word oder PowerPoint) erstellen und können relativ

leicht modifiziert werden. Schon deshalb sind sie den Nassi-Schneidermann Diagrammen vorzuziehen. Hinzu kommt der Vorteil, dass diese Programmpläne auch für einen nicht-Programmierer sehr leicht zu verstehen sind und an andere Ablaufdiagramme aus der UML<sup>1</sup> erinnern.

In der Abbildung ist wieder das gleiche Programm dargestellt.

<sup>1</sup> UML = Unified Modeling Language

# ÜBERSICHTSBLATT: PROGRAMME PLANEN

## Pseudocode

Die dritte Variante die wir hier behandeln wollen, ist der der sogenannte Pseudocode. Er unterliegt nicht annähernd so formalen Bestimmungen wie die beiden ersten Varianten. Im Prinzip geht man dabei so vor, dass man anfängt den Programmcode so wie man ihn sich vorstellt aufzuschreiben, dabei aber auf alle unnötigen Details verzichtet.

---

Programm:	Alle geraden Zahlen bis 99 ausgeben
Eingabe:	initialisierte Variable i
Ausgabe:	Alle geraden Zahlen bis 99 auf der Kommandozeile

---

```
do
    if (i mod 2 = 0) then
        gebe i aus
    fi
    erhöhe i um eins
while i < 100
```

---

Diese vereinfachte Form des Niederschreibens des Programmcodes, ohne dabei auf formale Eigenheiten der jeweiligen Sprache zu achten, erleichtert die Planung von komplizierten Algorithmen, da man sie quasi kochrezeptmäßig aufschreiben kann, und dann später nur die formalen Lücken füllen muss.