Auswahlstrukturen mit if .. else

Sie ist gekennzeichnet durch einen nicht linearen Ablauf mit einer Vorwärtsverzweigung. Der Ablauf gelangt an einen Entscheidungspunkt, an dem, abhängig von einer Bedingung, unterschiedliche Verarbeitungswege eingeschlagen werden. Das Entscheidungssymbol gibt eine Bedingung an (i. A. ein bedingter Ausdruck), deren Ergebnis ein Wahrheitswert ist.

# einseitige Auswahl

Diese Alternativstruktur führt nur auf einem der beiden Verzweigungspfade eine Anweisung(sfolge) aus und endet in der Zusammenführung beider Pfade.

## Syntax:

**if** (Bedingung) {

|  |  |
| --- | --- |
| Bedingung erfüllt?  Ja Nein | |
| Anweisung  **...** |  |

## Struktogramm:

Anweisung;

}

# Beispiel:

|  |  |
| --- | --- |
| alter < 0?  Ja Nein | |
| Ausgabe:  „Fehler“ |  |

**if** (alter < 0) { System.out.println (”Fehler“);

}

# zweiseitige Auswahl

Bei dieser Alternativstruktur führt jeder Verzweigungspfad auf jeweils eine eigene Anweisungsfolge. Sie endet auch wieder in einer Zusammenführung der Pfade.

## Syntax:

**if** (Bedingung) { Anweisung1;

|  |  |
| --- | --- |
| Bedingung erfüllt?  Ja Nein | |
| Anweisung1  **...** | Anweisung2  **…** |

## Struktogramm:

}

## else {

Anweisung2;

}

# Beispiel:

|  |  |
| --- | --- |
| a > b?  Ja Nein | |
| max  a | max  b |

**if** (a > b) { max = a;

}

## else {

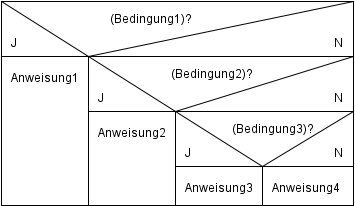
max = b;

}

# mehrseitige Auswahl

Bei der mehrseitigen Auswahl wird mit einer Folge von **else if** eine Kette von Abfragen aufgebaut.

## Syntax: Struktogramm

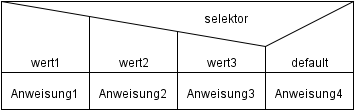
**if** (Bedingung1) { Anweisung1;

} **else if** (Bedingung2){   
 Anweisung2;  
} **else if** (Bedingung3){   
 Anweisung3;  
} else {  
 Anweisung4;  
}

Nur erste Anweisung, deren Bedingung erfüllt ist, wird ausgeführt. Ist keine Bedingung erfüllt, so wird, falls vorhanden, die letzte **else**-Anweisung ausgeführt. Als Anweisung kann auch eine Verbundanweisung mit einer Sequenz von Anweisungen in geschweiften Klammern (Block) ausgeführt werden.

# Fallunterscheidung mit switch .. case

## Syntax: Struktogramm

**switch** (selektor) {

**case** wert1:   
Anweisung1;   
**break**;

**case** wert2:   
Anweisung2;   
**break**;

**case** wert3:   
Anweisung3;   
**break**;

## default:

Anweisung4;

}

Hier erfolgt die Abfrage nicht nach einer Bedingung, sondern der bedingte Ausdruck (selektor) liefert einen konstanten Wert. Für jeden möglichen Ergebniswert ist ein Zweig vorgesehen.

Stimmt der Wert von selektor mit einem konstanten **case**-Ausdruck überein, werden alle dieser **case**-Zeile nachfolgenden Anweisungen bis zur **break**-Anweisung oder bis zum Ende des **switch**-Blocks ausgeführt. Die **break**-Anweisung sorgt dafür, dass der **switch**-Block verlassen wird. Existiert nicht für jeden möglichen Ergebniswert der Bedingung ein Pfad, ist ein zusätzlicher Pfad für alle nicht behandelten Fälle vorzusehen (**default**-Zweig).

Ab Java12 ist eine andere Syntax möglich, welche in komplexen Programmen, die lesbarkeit erhöhen kann, vgl. <https://www.infoq.com/articles/java-12-switch-expression/>