

## Programmiermethoden in der Mathematik WS 02/03

### Bedingte Anweisungen – Ergänzungen

Manchmal ist es sinnvoll, eine Schleife vorzeitig zu verlassen oder direkt zum nächsten Schleifendurchlauf zu springen. Dazu gibt es folgende Möglichkeiten:

- Die **continue-Anweisung**: Mit ihr wird in einer `for`, `while` oder `do while`-Schleife zum nächsten Schleifendurchlauf gesprungen. Die hinter `continue` stehenden Anweisungen werden *in diesem Durchlauf der Schleife* übersprungen. Beispiel:

```
while (A)
{
    B;
    if (C) continue;
    D;
}
```

Sobald in einem Schleifendurchlauf Bedingung `C` erfüllt ist, wird Anweisung `D` nicht mehr abgearbeitet, sondern es geht mit der direkt Überprüfung von `A` weiter.

- Mit der **break-Anweisung** wird die *nächste* übergeordnete Schleife direkt verlassen, ohne dass weitere Durchläufe stattfinden. Die Anwendung erfolgt analog zu `continue`.
- Mit der **exit-Funktion** wird das gesamte Programm direkt (und unsanft) beendet. Die Funktion erhält einen `int`-Parameter, der Möglichkeiten bietet, den Grund des Abbruchs nach außen zu geben (0 steht in Unix dabei für erfolgreichen Programmabschluss):

```
while (A)
{
    ...
    if (C) exit(1);
}
```

In der objektorientierten Programmierung vermeidet man diese Methode und fängt Fehler auf andere Weise ab.

Gibt es bei einer Bedingung mehr als zwei oder drei Möglichkeiten, so ist anstatt von verschachtelten `if-else`-Konstrukten die **switch-Anweisung** bequemer:

```
switch (I)
{
    case K1 : A1;
    case K2 : A2;
    case K3 : A3;
    default : A4;
}
```

`I` ist dabei ein Ausdruck, der einen *ganzzahligen Wert haben muss*. Da ein `char` ebenfalls eine ganzzahlige Darstellung hat (vgl. Typen), ist also auch ein Zeichen möglich. Meistens ist `I` ein Variablenname. Die `K1, K2, K3` sind Konstanten und `A1, A2, A3, A4` einzelne oder auch mehrere Anweisungen (die *nicht* geklammert werden müssen). Gilt `I==K1` so wird `A1` ausgeführt. Danach wird die `case`-Anweisung weiter durchlaufen und auch noch die anderen Alternativen werden überprüft. Am Ende wird die (optionale) `default`-Anweisung ausgeführt. Mit Hilfe eines `break` kann die `switch`-Anweisung verlassen werden, so dass nur jeweils eine Anweisung pro Fall ausgeführt wird:

```
switch (I)
{
    case K1 : A1; break;
    case K2 : A2; break;
    case K3 : A3; break;
    default : A4;
}
```

Damit ist die `switch`-Anweisung äquivalent zu entsprechend geschachtelten `if...else`-Anweisungen.