# Inkrement/Dekrement-Schreibweise

Wir wollen für alle Situationen,

in denen wir eine Befehlsfolge häufig benutzen,

in der Einsparungspotential besteht,

eine verkürzende Schreibweise definieren.

For-Schleifen brauchen wir häufig.

Wir analysieren,

wo wir eine Einsparung vornehmen können:

for (int i=1; i<=10; i = i+1) {…}

* Beim Schlüsselwort können wir nichts sparen
* auch nicht bei Deklaration u. Initialisierung,
* ebenso wenig bei der Bedingung.

Der Veränderungsteil lautet:

i = i + 1;

Dabei wird der Name der Variablen doppelt genannt.

Auch haben wir in den meisten Fällen eine Veränderung dahingehend,

dass wir um genau eins hoch- oder herunterzählen wollen.

Deshalb bietet es sich an,

hierfür eine abkürzende Schreibweise zu definieren,

bei der die involvierte Variable nur einmal erwähnt werden muss

u. die genau um 1 hoch- oder herunterzählt.

Das Fremdwort für "um 1 hochzählen" lautet "inkrementieren",

das für "um 1 herunterzählen" lautet "dekrementieren".

Daher die Bezeichnung der einzuführenden Schreibweise:

"Inkrement-Operator" u. "Dekrement-Operator".

Notwendige Infos dafür:

1. Variablenname, jedoch nur einmal erwähnt
2. Unterscheidung: Hoch- oder herunterzählen.

alte Schreibweise abkürzende Schreibweise

i = i + 1 ⬄ **i++**

Inkrement-Operatoren

oder **++i**

i = i - 1 ⬄ **i--**

Dekrement-Operatoren

oder **--i**

## Analyse des Benefits

alte Schreibeweise: neue Schreibweise:

i=i+1; ++i;

Benefit:

**1.** i=i+1; ++i;

Anzahl Ziffern: **123456 1234**

**Kürzer im Quelltext**

Ersparnis fällt umso mehr ins Gewicht, je länger der Variablenname ist.

2. Anzahl Befehle 1. i holen 1. i holen  
2. 1 holen -  
3. allgemeine Addition 2. Inkrement-Befehl  
 (relativ teuer) (relativ günstig)  
4. Ergebnis wegschaffen 3. Ergebnis weg

--------------------------------- -----------------------------

Summe: 4, darunter ein teurer Summe: 3, darunter ein billiger

schneller in der Ausführung

3. Übersichtlichkeit

Nehmen wir an, wir haben eine komplizierten Variablennamen: r4z8703

r4z8703 = r4z8703 + 1; r4z8703++;

Bei der altmodischen Schreibweise muss ich ziffernweise vergleichen,  
ob der Variablenname recht vom Gleichheitszeichen dem links entspricht.

Es wären beide Situationen denkbar:

1. Sie sind identisch: Die Variable soll um inkrementiert werden
2. Sie sind nicht identisch: Eine erste Variable soll ausgelesen werden  
   u. der um eins größere Wert soll in eine andere Variable geschrieben werden.

Dieser Vergleich entfällt bei der neuen Schreibweise.

Es ist auf den ersten Blick klar,  
dass nur eine Variable involviert ist,  
die inkrementiert werden soll.

Von daher ist die neue Schreibweise auch übersichtlicher  
(insbesondere bei komplexen Variablennamen).