# Pre-/Post - In- und Dekrementieren (++x, x++, --x, x--)

[*https://www.java-forum.org/thema/post-pre-inkrement-oder-dekrement.60468/*](https://www.java-forum.org/thema/post-pre-inkrement-oder-dekrement.60468/)

*Die genaue Semantik von i++ kann man sich so vorstellen:  
- t = i  
- i = i + 1  
- return t*

*Man merkt sich also den momentanen Wert in einer neuen, unsichtbaren Variable. Dann erhöht man i und gibt anschließend den Wert der unsichtbaren Variable zurück, also den Wert vor der Erhöhung.****Trotzdem ist i jetzt bereits erhöht! (aber ggf. nicht genutzt)***

*Nun zu Deinem eigentlichen Problem: array[i++] = i. Wir haben es hier mit einer Zuweisung zu tun, die durch den Zuweisungsoperator = realisiert ist. Bei allen binären Operatoren wird zuerst der linke Operand (hier array[i++]) und anschließend der rechte Operand (hier i) ausgewertet. Das kannst Du*[*hier*](http://java.sun.com/docs/books/jls/third_edition/html/expressions.html#15.7.1)*nachlesen.  
  
Da der Seiteneffekt von i++ unmittelbar beim Auswerten der linken Seite der Zuweisung eintritt, ist der Wert von i auf der rechten Seite der Zuweisung bereits um 1 höher.*

**daraus leite ich ab:**

Preinkrementierung:  
int i = 3;  
arr[++i] = i   
🡪 arr[4] = 4;

Postinkrementierung:  
int i = 3;  
arr[i++] = i;  
🡪 arr[3] = 4;

Predekrementierung:  
int i = 3;  
arr[--i] = i;   
🡪 arr[2] = 2;

Postdekrementierung:  
int i = 3;  
arr[i--] = i;   
🡪 arr[3] = 2;

<https://javabeginners.de/Grundlagen/Operatoren/Inkrement_und_Dekrement.php>

Der Unterschied zwischen Prä- und Postfix besteht in der Ausführungsreihenfolge. Beim Präfix wird es vor der Weiterverwendung, beim Postfix nach der Weiterverwendung der Variablen ausgeführt.

Die folgenden Multiplikations-Beispiele erläutern das Verhalten. Ihre Auswertung erfolgt von links nach rechts. In den Kommentierungen sind die jeweiligen Ergebnisse dargestellt.

## Inkrement als Postfix

123int i = 5, k = 5;

System.out.println(--i \* k++); // 20

System.out.println(i + ", " + k); // 4, 6

Ausführungsreihenfolge:

* verringere i um 1 dauerhaft (5 → 4)
* multipliziere i mit k (4 \* 5)
* erhöhe k um 1 dauerhaft (5 → 6)

## Inkrement als Präfix

123int i = 5, k = 5;

System.out.println(--i \* ++k); // 24

System.out.println(i + ", " + k); // 4, 6

Ausführungsreihenfolge:

* verringere i um 1 dauerhaft (5 → 4)
* erhöhe k um 1 dauerhaft (5 → 6)
* multipliziere i mit k (4 \* 6)

## Vorzeichen...

Die Inrement- und Dekrement-Operatoren dürfen nicht mit Vorzeichen-Operatoren verwechselt werden. Im Gegensatz zu jenen, führen diese keine Zuweisung durch, sodass sich der negative oder positive Wert nur auf die unmittelbar folgende Operation bezieht.

### ...mit Inkrement als Präfix

123int i = 5, k = 5;

System.out.println(i-- \* -++k); // -30

System.out.println(i + ", " + k); // 4, 6

Ausführungsreihenfolge:

* erhöhe k um 1 dauerhaft (5 → 6)
* wechsele das Vorzeichen von k für diese Operation (6 → -6)
* multipliziere i mit k (5 \* -6)
* verringere i um 1 dauerhaft (5 → 4)

### ...mit Inkrement als Postfix

123int i = 5, k = 5;

System.out.println(i \* -k++); // -25

System.out.println(i + ", " + k); // 5, 6

Ausführungsreihenfolge:

* wechsele das Vorzeichen von k für diese Operation (5 → -5)
* multipliziere i mit k (5 \* -5)
* erhöhe k um 1 dauerhaft (5 → 6)

Man beachte, dass das negative Vorzeichen bei der Ausgabe der Variablen nicht übernommen wird.

**Hier nochmal alle ohne Lösung zum Üben (Vorlage):**

int i = 5, k = 5;

System.out.println(--i \* k++); //

System.out.println(i + ", " + k); //

int i = 5, k = 5;

System.out.println(--i \* ++k); //

System.out.println(i + ", " + k); //

int i = 5, k = 5;

System.out.println(i-- \* -++k); //

System.out.println(i + ", " + k); //

int i = 5, k = 5;

System.out.println(i \* -k++); //

System.out.println(i + ", " + k); //

**Hier nochmal alle ohne Lösung zum Üben (Lösungsversuch):**

int i = 5, k = 5;

System.out.println(--i \* k++); // 20 (4\*5)

System.out.println(i + ", " + k); // 4, 5 -> 6

int i = 5, k = 5;

System.out.println(--i \* ++k); // 24 (4\*6)

System.out.println(i + ", " + k); // 4, 6

int i = 5, k = 5;

System.out.println(i-- \* -++k); // -30 (5\*-6)

System.out.println(i + ", " + k); // 4, 6

int i = 5, k = 5;

System.out.println(i \* -k++); // -25 (5 \* -5)

System.out.println(i + ", " + k); // 5, 6

Auflösung oben oder hier:  
<https://javabeginners.de/Grundlagen/Operatoren/Inkrement_und_Dekrement.php>