

## Einfache Programme in Java – Variablen und Methoden

**Variablen** werden für Berechnungen verwendet. Sie können verschiedene Werte annehmen und sind von einem bestimmten Typ:

Typ	Mögliche Werte und Größe
boolean	Nur zwei Möglichkeiten: true oder false (1 Bit)
char	Unicode-Zeichen z.B. A, a, Σ, φ, ä, ß... (16 Bit)
byte	Ganzzahl von -128 bis 127 (8 Bit)
short	Ganzzahl von -32768 bis 32767 (16 Bit)
int	Ganzzahl von -2147483648 bis 2147483647 (32 Bit)
float	Gleitkommazahl (32 Bit)
double	Gleitkommazahl doppelter Genauigkeit (64 Bit)

Erstelle ein neues Projekt mit dem Namen Variable und darin eine Klasse Taschenrechner. Du benötigst mehrere Variablen vom Typ Integer (Abkürzung int), die du im Konstruktor alle initialisieren solltest. Nehmen wir die Variable x mit dem Wert 0, y mit dem Wert 3, z mit dem Wert 5. Die Variable ergebnis setzen wir zunächst auf 0.

Schreibe eine Methode „addiereXZuY()“, die  $x + y$  errechnet, das Ergebnis in ergebnis speichert und mit dem Befehl `System.out.println(ergebnis);` ausgibt.

Schreibe eine Methode „subtrahiereYVonZ()“, die, na ja, du weißt schon was... . Auch hier soll das Ergebnis von Ergebnis mit dem oben genannten Befehl ausgegeben werden.

Schreibe eine weitere Methode „verdoppeleWertVonY“, in der du den Wert von y mal zwei nimmst ( $y=y*2$ , sprich“Die Variable y erhalte den Wert der Berechnung von  $y * 2$ “). Lasse dir den aktuellen Wert von y mit dem bekannten Befehl ausgeben.

Rufe danach alle drei Methoden auf, und zwar zweimal hintereinander und sage jeweils vor dem Aufruf das Ergebnis voraus.

**Zusammenfassung:** Variablen sind von einem bestimmten Typ, z.B. Integer (ganze Zahlen), double (Kommazahl), boolean (Wahrheitswert), char (Zeichen) etc. Sie können fast beliebig bezeichnet werden: x, willi, sdfhsadkjsa sind z.B. erlaubte Namen, es bieten sich aber sogenannte sprechende Namen an: ergebnis (für int oder double), teilgenommen (für boolean), A (für char) etc.

Die Wertzuweisung erfolgt über den Operator `=`, der nicht mit dem mathematischen „ist gleich“ verwechselt werden darf. Die Bedeutung ist vielmehr „erhält den Wert von“, also `x = x+y` bedeutet „Die Variable x erhält als neuen Wert das Ergebnis der Berechnung von  $x+y$ “.

## Methoden

Es gibt verschiedene Arten von Methoden:

- die sondierende Methode, die einen Wert abfragt
- die verändernde Methode, die einen Wert ändert

Füge eine Methode `zeigeErgebnis()` an, die den Wert von `ergebnis` ausgibt, nachdem sie ihn „sondiert“ hat. Der Befehl lautet `System.out.println(„Der Wert der Variable ergebnis ist“ + ergebnis);` Das `+` hängt den Wert der Variablen an den Text an.

Versuche eine alternative Methode:

```
public int gibErgebnis()
{
    return ergebnis;
}
```

Dir ist aufgefallen, dass die Methode kein `void` enthält. `void` bedeutet ja, dass kein Rückgabewert erwartet wird, hier wird aber der Wert der Variable `ergebnis` erwartet, deswegen kein `void`, dafür aber der Typ des Ergebnisses. Da wir wissen, dass `ergebnis` vom Typ `Integer` ist, sollte das also keine Überraschung sein.

Nun zu den verändernden Methoden: Du kannst den Wert einer Variablen ändern, ohne dass das Ergebnis zurückgemeldet wird, z.B. mit der oben benutzten Methode `addiereXZuY()`. Die Methode sieht wie folgt aus:

```
public void addiereXZuY()
{
    ergebnis=x+y;
}
```

Der Nachteil einer solchen Methode ist, dass der Nutzer keinen Einfluss auf die Berechnung nehmen kann. Soll der Nutzer einen Wert eingeben können, wird der Variablentyp gefolgt vom Namen der Variablen angegeben. Diese Variable muss vorher nicht deklariert sein. Beispiel:

```
public void neuerXWert(int neuesX)
{
    x = neuesX;
}
```

Erklärung: `void` heißt, es wird kein Wert zurückerwartet, `neuerXWert` ist der Methodename, `int neuesX` bedeutet, dass der Nutzer einen Wert vom Typ `Integer` eingibt und dieser Wert unter dem Variablennamen `neuesX` verwendet wird. Außer an dieser Stelle muss der Variablenname `neuesX` nirgendwo deklariert werden. Wird die Methode aufgerufen, wird der bisherige Wert von `x` mit dem Wert, den der Nutzer eingegeben hat, überschrieben. `x` muss allerdings sehr wohl vorher deklariert worden sein.

### Aufgaben:

- a) Schreibe eine Methode `neuerErgebniswert()`, die dem Nutzer erlaubt, einen beliebigen neuen Wert für `ergebnis` zu setzen!
- b) Jetzt endlich zum Taschenrechner: Schreibe eine Methode mit dem Namen `initialisiereStartzahl`, die dem Nutzer erlaubt, der Variable `zahl` einen Startwert zuzuschreiben. Schreibe danach eine Methode `addiere`, bei der der Nutzer eingeben kann, welchen Wert zu `zahl` addiert werden soll. Führe die Addition aus und gib das Ergebnis zurück (`System.out.println(„Das Ergebnis der Addition von“ + zahl + „ plus „ + eingegebene Zahl + „ ist „ + ergebnis);`
- c) Nun zu einem Automaten: Schreibe eine Klasse `BloederTicketautomat`. Dieser soll

- auf Anfrage den von dir festzusetzenden Ticketpreis angeben
- dem Nutzer die Möglichkeit geben Geld „einzuwerfen“
- eine Abfrage erlauben, wie viel Geld bisher eingeworfen worden ist
- und ein Ticket „ausgeben“, das soll dann wie in den nächsten drei Zeilen aussehen  
`(System.out.println(„#####“);`  
`(System.out.println(„ Ihr Ticket!“);`  
`(System.out.println(„#####“);`

Wähle „sprechende“ Variablennamen und ignoriere die Notwendigkeit vor der Ticketausgabe zu prüfen, ob überhaupt genügend bezahlt worden ist.