

Kleine Übersicht über bislang kennengelernte Begriffe in Java

Schlüsselwörter / reservierte Wörter

In Java gibt es verschiedene Schlüsselwörter bzw. reservierte Wörter.

Diese haben verschiedene Funktionen, welche von vornherein festgelegt sind.

Sie haben eine bestimmte Bedeutung und können nicht, wie z.B. Variablen oder Funktionsnamen neu festgelegt bzw. verändert werden, ebenso darf man Variablen, Klassen oder Methoden nicht so benennen. Sie sind in Entwicklungsumgebungen oft farblich markiert. Es gibt in Java ungefähr 50 reservierte Begriffe. Außerdem sind sie immer in Kleinbuchstaben geschrieben.

Manche Schlüsselwörter werden (noch) nicht verwendet, sind aber trotzdem reserviert.

Beispiele:

if, while, for, new, return, public, static, void, usw.

Im erweiterten Sinne kann man auch die Operatoren dazuzählen, deren Bedeutung man ebenfalls nicht ändern kann. Ebenso darf man z.B. keiner Variablen den Namen "&&" geben.

Operatoren: == (Vergleich), = (Wertzuweisung), && ("und"), || ("oder"),

<, >, <=, >=, != (ungleich), +, - *, /, ! ("nicht"), %(Divisionsrest),

"." (um Methoden auf Objekten oder Klassen aufzurufen)

++, -- (Variable um 1 erhöhen, erniedrigen)

Methoden (aus einer bestimmten Klasse/Interface)

Methoden sind in der objektorientierten Programmierung Unterprogramme (in der Form von Funktionen oder Prozeduren), die das Verhalten von Objekten beschreiben und implementieren.

Über die Methoden des Objekts können Objekte untereinander in Verbindung treten.

Jedes Java-Programm muss an einer Stelle beginnen. Dies ist immer eine `main`-Methode. Klassen, die eine `main`-Methode, engl. für ‚Hauptmethode‘, enthalten, sind ausführbare Klassen.

Beispiele:

`setFont` (ist Methode von `JComponent`), `actionPerformed` (aus Interface `ActionListener`),

`equals` (Methode der Klasse `String`)

`random` (Methode der Klasse `Math`)

`paint` (Methode der Klasse `Component`)

Standard-Java-Klassen für GUI-Erstellung, die wir ggf. importiert haben:

`JButton`, `JLabel`, `JFrame`, `JPanel` etc. sind praktisch vorgeschriebene Klassen, die man in sein Programm importieren kann, um sich zusätzliche Arbeit zu sparen. Sie helfen bei der Erstellung grafischer Benutzeroberflächen um z.B Knöpfe, Textfelder usw. darzustellen.

Diese Klassen kann man durch z.B. `import javax.swing.JComponent` in eigene Klassen importieren, damit man sie nicht erneut schreiben / erklären muss.

Zwar kann man ein `JButton`-Objekt erstellen, um ein Button im Fenster darzustellen. Jedoch muss man dann immer noch seine Funktion programmieren.

atomare Datentypen

In Java wird zwischen primitiven (oder atomare) Datentypen und Referenzdatentypen unterschieden. Atomare Datentypen sind eine Gruppe von Typen, mit denen man Variablen erstellen kann, die Zahlen, einzelne Zeichen oder logische Werte haben. (Referenzdatentypen beziehen sich auf Objekte)

Beispiele:

`integer`, `long` (ganze Zahlen)

`boolean` (nur `true` oder `false`) – oft bei **if** / **while** verwendet: "*Bedingung erfüllt*" – oder eben nicht.

`double`, `float` (Kommazahlen)

`char` (ein einziger Buchstabe)

selbstgewählte Methodennamen

Prinzipiell ist man bei der Wahl von Methodennamen ziemlich frei. Zunächst gibt es aber die Konvention: Ein Methodename beginnt mit einem Kleinbuchstaben. Kommen im Methodennamen mehrere Wörter vor, so werden die Leerzeichen ausgelassen und die Anfangsbuchstaben der weiteren Wörter groß geschrieben (z.B. `getCurrentTimeMillis()`)

Oft verwendet man sogenannte "Getter" und "Setter"-Methoden, um Objekteigenschaften zu beeinflussen. Hat ein Objekt z.B. eine Position in Form von x- und y-Koordinaten, so ist es üblich, die Methoden "getX" und "setX" bzw. "getY" und "setY" bereitzustellen.

Beispiele:

`get` (oft als Teil eines Methodennamens z.B. "getPosition" um einen bestimmten Wert zu erfahren)

`setPosition` (selbstgeschriebene Methode aus z.B. dem Smiley-Projekt)

selbstgewählte Objektnamen (oder auch Variablennamen)

Bei Variablen gelten die gleichen Namenskonventionen wie bei Methoden (mit Kleinbuchstaben beginnend, ggf. Großbuchstaben bei aneinandergereihten Worten).

Beispiele:

`but` (oft als Name eines Objekts der Klasse `JButton`)