

Grundelemente eines Java Programms

((itanius informatik))

Java Foundation Track by Carsten Bokeloh

Grundelemente eines Java Programms

- Kommentare
- Bezeichner und Namen
- Literale
- Trennzeichen
- Interpunktionszeichen
- Operatorsymbole
- import-Anweisungen

Kommentare

- ❖ Zeilenkommentar - > **//hier ein Kommentar**
- ❖ Blockkommentar -> **/* Dies ist ein Blockkommentar */**
- ❖ JavaDoc - Kommentar

/**

*** @author Carsten Bokeloh**

*** @version 1.0**

***/**

Bezeichner und Namen

erlaubt sind:

- ❖ starten mit einem Zeichen, Währungssymbol oder Verbindungszeichen
- ❖ starten nicht mit einer Nummer
- ❖ können nach dem ersten Zeichen alle Kombinationen von Zeichen, Nummern, Währungssymbolen und Verbindungszeichen enthalten
- ❖ theoretisch ist die Anzahl der Zeichen pro Identifier unbegrenzt
- ❖ keine Schlüsselworte wie new
- ❖ Identifier sind case-sensitive, Groß- und Kleinschreibung wird unterschieden

<code>abstract</code>	<code>double</code>	<code>int</code>	<code>super</code>
<code>assert</code>	<code>else</code>	<code>interface</code>	<code>strictfp</code>
<code>boolean</code>	<code>extends</code>	<code>long</code>	<code>switch</code>
<code>break</code>	<code>false</code>	<code>native</code>	<code>synchronized</code>
<code>byte</code>	<code>final</code>	<code>new</code>	<code>this</code>
<code>case</code>	<code>finally</code>	<code>null</code>	<code>throw</code>
<code>catch</code>	<code>float</code>	<code>package</code>	<code>throws</code>
<code>char</code>	<code>for</code>	<code>private</code>	<code>transient</code>
<code>class</code>	<code>goto</code> ●	<code>protected</code>	<code>true</code>
<code>const</code> ●	<code>if</code>	<code>public</code>	<code>try</code>
<code>continue</code>	<code>implements</code>	<code>return</code>	<code>void</code>
<code>default</code>	<code>import</code>	<code>short</code>	<code>volatile</code>
<code>do</code>	<code>instanceof</code>	<code>static</code>	<code>while</code>

Literale

- ❖ ganze Zahlen
- ❖ Gleitkommazahlen
- ❖ Wahrheitswerte
- ❖ einzelne Zeichen in einfachen Hochkommata ('a')
- ❖ Zeichenketten in Anführungszeichen ("Hallo")
- ❖ das sogenannte null-Literal (null)

Trennzeichen

- ❖ Leerzeilen
- ❖ Zeilenendezeichen(Enter Taste)
- ❖ Tabulatorzeichen
- ❖ Kommentare
- ❖ Operatoren(+ oder *)
- ❖ die Interpunktionszeichen . , ;) ({ } []

Interpunktionszeichen

- ❖ anstatt eines Punktes wie in der deutschen Sprache benutzt Java das Semikolon ;
- ❖ Befehle werden immer mit einem Semikolon abgeschlossen
- ❖ Fehlen Sie, meckert der Compiler

Operatorsymbole

- ❖ Operatoren, die nur einen Operanden benötigen
 - ❖ `++` oder `--`
- ❖ Operatoren, die zwei Operanden verknüpfen
 - ❖ `+` `-` oder `==`
- ❖ Verknüpfung von drei Operanden
 - ❖ `?:` (Kurzform für if else)

import Anweisungen

- ❖ `import de.lufthansa.cargo.*;`
- ❖ `import Prog1Tools.IOTools;`

Im Buch gibt es mehr Infos unter Kapitel 2.1

[http://openbook.galileocomputing.de/javainsel/
javainsel_02_001.html#dodtpb2fd0fcf-b156-46cc-b527-3b1c995ceda2](http://openbook.galileocomputing.de/javainsel/javainsel_02_001.html#dodtpb2fd0fcf-b156-46cc-b527-3b1c995ceda2)

Übung

Welche Bezeichner sind zulässig?

Peter der Grosse

Peter_der_Grosse

Peter,der_Grosse

0 Ahnung?

0_Ahnung

null_Ahnung!

1234abc

_1234abc

_1_2_3_4_abc

Lösung

Peter der Grosse

Leerzeichen innerhalb eines Bezeichners sind nicht erlaubt

Peter_der_Grosse

Der Unterstrich (Underscore) in Bezeichnernamen ist zulässig

Peter,der_Grosse

Das Komma ist nicht erlaubt, bei einer Deklaration würde der Compiler dies als zwei (durch ein Komma getrennte) Bezeichner auffassen (z.B. *int Karl,der_Grosse*), nämlich *Karl* und *der_Grosse*

Lösung

0 Ahnung?

Bezeichner dürfen weder mit Ziffern beginnen noch dürfen sie ein ? enthalten

0_Ahnung

Bezeichner dürfen nicht mit Ziffern beginnen

null_Ahnung!

Auch ein Ausrufezeichen im Bezeichner-Namen ist nicht gestattet.

Lösung

1234abc

Bezeichner dürfen nicht mit Ziffern beginnen

_1234abc

Der Unterstrich (Underscore) als erstes Zeichen eines Bezeichners ist gestattet (einige Autoren empfehlen sogar, Bezeichner für globale Variablen

Bringen Sie folgendes Programm zum Laufen

```
/* Beispiel : Berechnung von 49 + 81 */
public class Berechnung {
    public static void main (String args) {
        {
            int summe;
            summe = 49 + 82;
            System.out.println("49 + 81 ergibt");
            System.out.println(summe);
        }
    }
}
```