

Grundelemente von Java

- Zeichensatz
- Kommentare
- Token
- Bezeichner
- Schlüsselwörter
- Einfache Typen
- Literale
 - Ganze Zahlen
 - Reelle Zahlen
 - Wahrheitswerte
 - Zeichen
 - Zeichenketten
- Variablen und Variablen-deklarationen
- Benannte Konstanten

Grundelemente von Java

Zeichensatz

bekannte Zeichensätze:

- **EBCDIC (Extended Binary Coded Decimal Interchange Code)**
8-Bit-Code,
Verwendung im Großrechnerbereich
- **ASCII (American Standard Code for Infomation Interchange)**
7-Bit-Code, d. h. 128 Zeichen
- **ISO-8859-1 Latin-1**
Erweiterung des ASCII-Zeichensatzes auf
8 Bit, d. h. 256 Zeichen
- **Unicode**
16-Bit-Zeichensatz,
d. h. 65536 verschiedene Zeichen darstellbar

Java verwendet Unicode für die interne Darstellung von Zeichen.

Die externe Darstellung von Zeichen hängt vom zugrundeliegenden Betriebssystem ab.

Grundelemente von Java

Kommentare

drei Arten von Kommentaren:

```
// Dies ist ein Kommentar wie in C++.
```

Alle Zeichen von // an bis zum Zeilenende werden ignoriert.

```
/* Dies ist ein Kommentar wie in C. */
```

Alle Zeichen zwischen /* und dem nächsten */ werden ignoriert.

```
/** Dies ist ein spezieller Kommentar für javadoc. */
```

Alle Zeichen zwischen /** und */ werden ignoriert.

Diese Kommentare kennzeichnen Text, der von javadoc übernommen wird, einem automatischen Dokumentationsgenerator.

Grundelemente von Java

Kommentare

Beispiel: javadoc-Kommentare

Java-Code:

```
public class JavaDoc
{
    /**
     * hierher wird verwiesen
     */
    public void dummy()    { }

    /**
     * Formatierte Ausgabe eines Strings.
     * @see JavaDoc#dummy
     * @param s auszugebender String
     * @param breite Ausgabebreite (Anzahl
     * auszugebender Zeichen)
     * @param fuell Füllzeichen
     * @return formatierter String
     */
    public static String Format
        (String s, int breite, char fuell)
    {
        String erg = new String();
        // irgendwas
        return erg;
    }
}
```

Grundelemente von Java

Kommentare

Beispiel: javadoc-Kommentare

Ansicht des erzeugten HTML-Textes (Ausschnitt):

The screenshot shows a Microsoft Internet Explorer window displaying JavaDoc-generated HTML. The title bar reads "Generated Documentation (Untitled) - Microsoft Internet Explorer". The left sidebar has links for "All Classes" and "JavaDoc". The main content area is titled "Constructor Detail" and contains the JavaDoc for the `JavaDoc` class. It shows the constructor `public JavaDoc()`. Below this is a "Method Detail" section for the `dummy` method, which is described as returning a formatted string. The `Format` method is also shown, along with its parameters (`s`, `breite`, `fuell`) and return type (a formatted string). There are also sections for "Parameters", "Returns", and "See Also". At the bottom, there are navigation links for "Class", "Tree", "Deprecated", "Index", and "Help", and links for "PREV CLASS", "NEXT CLASS", "SUMMARY", "INNER", "FIELD", "CONSTR", and "METHOD". On the right side, there are links for "FRAMES", "NO FRAMES", "DETAIL", "FIELD", "CONSTR", and "METHOD". The status bar at the bottom shows various open tabs and the time "18:11".

Grundelemente von Java

Kommentare

Warum kann der Java-Compiler das folgende Beispiel nicht übersetzen?

```
/* zur Zeit auskommentiert,  
da noch nicht implementiert  
  
/* ein paar echt coole Dinge berechnen */  
universum.cooleDinge();  
  
*/
```

Grundelemente von Java

Token

Die **Token** einer Sprache, auch

- **Symbole** oder
- **lexikalische Einheiten**

genannt, sind die Wörter, auf denen die Sprache basiert.

Beispiele für Token:

- Bezeichner
- Schlüsselwörter
- Operatoren

Wortzwischenräume wie

- Leerzeichen
- Tabulatoren
- Zeilenvorschub-Zeichen
- Seitenvorschub-Zeichen

haben keine Bedeutung, außer dass sie die Token voneinander trennen.

Es sei denn, sie sind in Zeichen- oder Zeichenketten-Literalen enthalten.

Grundelemente von Java

Bezeichner

Bezeichner (vom Programmierer verwendete Namen) dürfen beliebig lang sein.

Bezeichner müssen beginnen mit einem

- Buchstaben,
- Unterstrich (_) oder
- Dollarzeichen (\$).

Die folgenden Zeichen können auch Ziffern sein.

Groß-/Kleinschreibung wird unterschieden.

Beispiele:

```
i  
kalorien  
$_99  
_read  
Häagen_Dazs  
PI  
pi  
Pi  
Begrüßung  
año  
dépannage
```

Grundelemente von Java

Bezeichner

Welche der folgenden Namen sind gültige Java-Bezeichner?

count
n1
salesTax
sales_tax
hoch-2
DaysOfWeek
zehn%
_code
split Word
\$monthlyIncome
MinutesIn365DAYS
minutesIn365Days
minutes_in_365_days
leapYear
5_Gang_Getriebe
if
class
SPEED
customerlastname

Grundelemente von Java

Schlüsselwörter

Schlüsselwörter sind reservierte Begriffe und dürfen nicht als Bezeichner verwendet werden.

Verwendet für Typenamen

boolean
char
byte short int long
float double
void

Verwendet für Ausdrücke

new this super
null true false

Verwendet in Anweisungen

Auswahlanweisungen
if else
switch case break default

Wiederholungsanweisungen
for continue
do while

Grundelemente von Java

Schlüsselwörter

Verwendet in Anweisungen

Anweisungen zur Übertragungssteuerung

return

throw

Überwachungssysteme

(threads, Ausnahmebehandlung, Zusicherungen)

synchronized

try catch finally

assert

Verwendet zur Modifikation von Deklarationen

static

abstract final strictfp

private protected public

transient volatile

Verwendet für Methoden oder Klassen

class instanceof

throws native

Grundelemente von Java

Verwendet für Komponenten, die größer als Klassen sind

extends
interface implements
package import

Zur Zeit unbenutzt, aber ebenfalls reserviert

const goto

Grundelemente von Java

Einfache Typen

Typ	Inhalt	Standard	Größe	Minimalwert	Maximalwert
boolean	true oder false	false	1 Bit	-	-
char	Unicode-Zeichen	\u0000	16 Bit	\u0000	\uFFFF
byte	Integer mit Vorzeichen	0	8 Bit	-128	127
short	Integer mit Vorzeichen	0	16 Bit	-32768	32767
int	Integer mit Vorzeichen	0	32 Bit	-2147483648	2147483647
long	Integer mit Vorzeichen	0	64 Bit	-9223372036854775808 9223372036854775807	
float	IEEE 754 Fließpunktzahl	0.0	32 Bit	$\pm 1.40239846 \times 10^{-45}$ $\pm 3.40282347 \times 10^{38}$	
double	IEEE 754 Fließpunktzahl	0.0	64 Bit	$\pm 4.94065645841246544 \times 10^{-324}$ $\pm 1.79769313486231570 \times 10^{308}$	

Grundelemente von Java

Literale

In Java besitzt jeder Typ Literale, die für die Konstanten des Typs stehen.

Ganze Zahlen (byte, short, int, long)

Ganzzahl-Literale werden durch Zeichenketten beschrieben, die aus oktalnen, dezimalen oder hexadezimalen Ziffern bestehen.

Welche Basis eine Zahl hat, wird durch die vorderen Zeichen bestimmt:

- führende 0 (Null)
=> Oktalzahl (Basis 8)
- 0x oder 0X am Anfang
=> Hexadezimalzahl (Basis 16)
- alle anderen Ziffern
=> Dezimalzahl (Basis 10)

Die folgenden Zahlen haben alle den gleichen Wert:

29 035 0x1D 0X1d

Ganzzahl-Literale sind vom Typ `int`, es sei denn, man fügt ein kleines oder großes `'L'` an, dann sind sie vom Typ `long`.

Grundelemente von Java

Literale

Reelle Zahlen (float, double)

Gleitkommazahlen werden durch Dezimalzahlen mit optionalem Dezimalpunkt dargestellt, denen ein Exponent folgen kann. Es muss mindestens eine Ziffer vorhanden sein.

Die folgenden Literale beschreiben alle die gleiche reelle Zahl:

18. 1 . 8e1 .18E2

Gleitkomma-Literale sind vom Typ double, es sei denn, sie sind durch ein kleines oder großes ,F' als Konstante vom Typ float gekennzeichnet.
Ein angehängtes kleines oder großes ,D' spezifiziert eine Konstante vom Typ double.

Wahrheitswerte (boolean)

true und false

werden von Vergleichen wie z. B. < zurückgeliefert

Grundelemente von Java

Literale

Zeichen (char)

Zeichen-Literale werden zwischen zwei Apostrophen (auch Hochkomma genannt) geschrieben; z. B. 'H'.

Einige Sonderzeichen können durch spezielle Ersatzdarstellungen (Escape-Sequenzen) ausgedrückt werden.

Ersatzdarstellung	Zeichen	Wert	Erläuterung
\b	BS	\u0008	Backspace (Zeichen zurück)
\t	HT	\u0009	(horizontaler) Tabulator
\n	LF	\u000a	Zeilenvorschub (engl. <i>newline</i> oder <i>line feed</i>)
\f	FF	\u000c	Seitenvorschub (engl. <i>form feed</i>)
\r	CR	\u000d	Wagenrücklauf (engl. <i>carriage return</i>)
\"	"	\u0022	Anführungszeichen oben (engl. <i>double quote</i>)
'	'	\u0027	Apostroph (engl. <i>single quote</i>)
\\	\	\u005c	inverser Schrägstrich (engl. <i>backslash</i>)
\ddd			Zeichen mit dem oktalen ASCII-Wert ddd (0000 <= ddd <= 0377)
\xxxxx			Unicode-Zeichen mit der hexadezimalen Codierung xxxx (0x0000 <= xxxx <= 0xffff)

Grundelemente von Java

Literale

Zeichenketten (String)

Zeichenketten-Literale sind Folgen von Zeichen-Konstanten, die zwischen zwei Anführungszeichen ("") stehen.

Zeichenketten dürfen auch Ersatzdarstellungen aus der obigen Tabelle enthalten. Sie können mit dem Operator + aneinandergehängt (konkateniert) werden.

Beispiel:

Die Anweisung

```
System.out.print( "Hallo Leute, \nwie "
                  + "geht es Euch?\n" );
```

liefert die folgende Ausgabe auf den Bildschirm:

```
Hallo Leute,
wie geht es Euch?
```

Grundelemente von Java

Variablen und Variablendeklarationen

Eine **Variable** benennt einen bestimmten Speicherbereich. Eine Variable besteht aus den folgenden drei Angaben:

- Name radius
- Datentyp float
- Wert 17.5e3F

Der **Name** ist ein Bezeichner und identifiziert die Speicheradresse.

Der **Datentyp** legt fest, in welchem Wertebereich der Wert der Variablen liegen kann.

Der **Wert** der Variablen ist der momentane Inhalt des Speicherbereichs.

Grundelemente von Java

Variablen und Variablen-deklarationen

Variablen müssen vor ihrer Verwendung deklariert werden. Sie können bei der Deklaration initialisiert werden.

Syntax der Variablen-deklaration (vereinfacht):

Modifizierer-Liste Typname Bezeichner-Liste ;

Bezeichner-Liste ist eine Folge von Bezeichnern, die durch Komma getrennt sind. Für jeden Bezeichner kann ein Initialisierungsausdruck angegeben werden.

Beispiele:

```
int    daysInMonth, daysInYear = 365;  
short  smallInt;  
long   bigInt;  
int    one = 1, two = 2, three;
```

Grundelemente von Java

Variablen und Variablendeklarationen

Beispiel: Das Programm

```
public class EinfacheKlasse
{
    public static void main(String[] argv)
    {
        float f1 = 12.34e5F;
        long lg1 = 4321567;
        char ch = 'J';
        String s =
            "Moechten Sie die Zahlenwerte sehen? ";
        System.out.println(s);
        System.out.println(
            "Die Antwort ist (J/N) : " + ch);
        System.out.println("f1 = " + f1);
        System.out.println("lg1 = " + lg1);
    }
}
```

erzeugt die folgende Ausgabe:

```
Moechten Sie die Zahlenwerte sehen?  
Die Antwort ist (J/N) : J  
f1 = 1234000.0  
lg1 = 4321567
```

Grundelemente von Java

Benannte Konstanten

Häufig verwendete Konstanten kann man benennen und dann im Programm diesen Bezeichner verwenden. Dies erhöht die Lesbarkeit und Flexibilität des Programms.

Zur Definition von benannten Konstanten wird der Modifizierer `final` benutzt.

Die Konstante PI kann folgendermaßen definiert werden:

```
final double PI = 3.141592654;
```

Der Wert von `PI` kann im Programm nicht verändert werden. Die Anweisung

```
PI = 0.05;
```

führt zu einer Fehlermeldung.