Einstieg in die Informatik mit Java

Grundlagen

Gerd Bohlender

Institut für Angewandte und Numerische Mathematik

Gliederung

Gliederung

Kommentare

```
// Kommentar bis zum Zeilenende

/* Kommentar evtl. ueber ...
   mehrere ...
   Zeilen
*/

/** Kommentar evtl. ueber mehrere Zeilen
   (Dokumentationskommentare) */
```

Kommentare – Schachtelung

Achtung

Kommentare können nicht beliebig geschachtelt werden und dürfen nicht in Namen und Wortsymbolen stehen!

Fehlerhafte Schachtelung

```
/*
    a = b + c;                  /* Addition */
    System.out.println(a);      /* Ausgabe  */
*/
```

Mögliche Schachtelung

```
/*
    a = b + c;                  // Addition
    System.out.println(a);      // Ausgabe
*/
```

Kommentare – Dokumentationskommentare

Dokumentationskommentare werden wirksam, wenn man die Java–Datei mit dem im JDK enthaltenen Programm `javadoc` bearbeitet. `javadoc` legt mehrere `html`–Dateien an, die zusammen mit den `class`–Dateien als Dokumentation ausgeliefert werden können. Folgende „Tags“ werden dabei berücksichtigt:

Tag	**Bedeutung**	**Anwendung**
`@see`	Verweis auf andere Stelle	Klasse, Methode, Variable
`@author`	Name des Autors	Klasse
`@version`	Versionsnummer	Klasse
`@param`	Name u. Beschreibung von Parametern	Methode
`@return`	Beschreibung des Funktionswertes	Methode
`@exception`	Name u. Beschreibung von Ausnahmen	Methode

Gliederung

Bezeichner für Klassen, Methoden, Variablen

- *Bezeichner* (auch geläufig als *Namen*) bestehen aus beliebig langen Folgen von Unicodebuchstaben und Ziffern.
- Ein Bezeichner beginnt mit einem Buchstaben, es folgen Buchstaben oder Ziffern.
- `A` bis `Z`, `a` bis `z`, `_` und `$` sind Unicodebuchstaben, `0` bis `9` sind Unicodeziffern.
- Nicht erlaubt sind alle Wortsymbole und die Literale `true`, `false` und `null`.

Bezeichner – Beispiele

Beispiel

`Anfang`, `Ende`, `x1`, `_425`

Achtung

Java unterscheidet in der Groß– und Kleinschreibung! Dadurch sind folgende Bezeichner erlaubt: `Int`, `INT`
Dagegen verboten, da ein reserviertes Wortsymbol: `int`

Konventionen
Für die Bezeichner von Klassen, Methoden, Variablen und Konstanten wurden die folgenden Vereinbarungen getroffen:

- Klassen: Anfangsbuchstabe groß, Rest klein, Wortanfänge groß, z. B. `HalloWelt`,
- Methoden und Variablen: Anfangsbuchstabe klein, sonst identisch mit Konventionen für Klassennamen, z. B. `main`, `xWert`, `piMalDaumen`,
- Konstanten: lauter Grossbuchstaben, z. B. `MAX`, `DIM`.

Gliederung

White–Space–Zeichen

Unter dem Begriff *White–Space–Zeichen* werden folgende Zeichen zusammengefaßt:

- Leerzeichen,
- Zeilenendezeichen,
- Tabulator,
- Seitenvorschub.

White–Space–Zeichen erhöhen die Lesbarkeit des Quelltextes und werden zum Trennen von Wortsymbolen, Bezeichnern, usw. benötigt (z. Bsp. `public class Hallo`). Ansonsten besitzen sie keine Wirkung.

Achtung

White–Space–Zeichen dürfen nicht innerhalb von Namen und Wortsymbolen stehen!

Gliederung

Wortsymbole

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

Achtung

Die Literalkonstanten `true`, `false` und `null` dürfen ebenfalls nicht als Bezeichner verwendet werden!

Gliederung

Interpunktionszeichen

( ) { } [ ] ; , .

Gliederung

Operatoren

Insgesamt benutzt Java 36 Operatoren:

=	<	>	!=	? :	==
>>>=	<=	>=	&&	++	--
+	-	*	/	|	^
%	<<	>>	+=	-=	*=
/=	& =	|=	^=	%=	<<=
>>=	!	>>>	&	||	~

Gliederung

import–Anweisungen

`import`–Anweisungen werden verwendet, um Bibliotheken, die in separaten Klassen ausgelagert sind, der eigenen Klasse zugänglich zu machen.

Beispiel

```
import java.util.*;    für die Eingabe,
import java.awt.*;     für graphische Elemente.
```

Achtung

`import`–Anweisungen müssen vor der Definition der Klasse erfolgen!

Gliederung

Form eines Programms

Wesentliches Konzept von Java ist die objektorientierte Programmierung. Daher muss die `main`–Methode in einer Klasse definiert sein.

Für Applikationen muss genau eine Methode der Form

```
public static void main (String[] args)
```

definiert sein, dies ist das Hauptprogramm.

Typischer Aufbau eines Programms

```
import–Anweisungen
public class Klassenname {
        Definitionen globaler Variablen und Methoden
        public static void main (String[] args){
                Definition lokaler Variablen
                Anweisungen
        }
        weitere Definitionen
}
```

Typische Anweisungen sind Wertzuweisungen der Form *Variable = Ausdruck* und die Ausgabe von Zeichenketten auf dem Bildschirm.