

JAVA

Kommentieren mit Javadoc

Kommentare in Java

```
/* mehrzeiliger Kommentar */
```

```
// einzeiliger Kommentar
```

- Sichtbar nur im Quelltext
- Beschreiben für Entwickler was und wie etwas implementiert wurde
 - ◆ Programme alleine sind für Menschen nicht selbsterklärend
- Entwickler
 - ◆ Teamkollegen, die Quelltext verstehen und ändern müssen
 - ◆ Man selbst, um eigenen Quelltext nach Wochen oder Jahren wieder zu verstehen

```
/**
```

```
 * Javadoc Kommentar vor Klassen, Methoden, Attributen
```

```
 */
```

- Sichtbar im Quelltext und als HTML Dokumentation
- Beschreibt wofür Klasse verantwortlich ist und was eine Methode macht (aber nicht wie eine Methode etwas macht)
- Optionaler erster * in jeder Zeile innerhalb des Kommentars wird von Javadoc ignoriert
- Auch für Entwickler, die Quelltext nicht zur Verfügung haben
 - ◆ Z.B. Java Klassen des JDK

Javadoc Kommentare

```
/**  
 * Geschwindigkeit ist verantwortlich für die Umrechnung  
 * verschiedener Geschwindigkeitsmasse untereinander.  
 * Für bestmögliche Genauigkeit wird bei jeder Umrechnung  
 * ausschließlich double verwendet.  
 */  
public class Geschwindigkeit {  
    // Methoden, ...  
}
```

- Javadoc Kommentar **vor** Klassendeklaration
- *Immer* ein Satz, der *kurz* beschreibt, für was die Klasse entworfen und implementiert wurde
- Weitere Sätze, falls nötig, die mehr Details beschreiben.

Javadoc Tags

```
/**  
 * Geschwindigkeit ist verantwortlich für die Umrechnung  
 * verschiedener Geschwindigkeitsmasse untereinander.  
 *  
 * @author Christian Pape  
 *  
 * @version 1.0  
 */
```

```
public class Geschwindigkeit {  
    // Methoden, ...  
}
```

- Javadoc kennt verschiedene „Tags“, mit denen Zusatzinformationen angegeben werden können.
- Diese werden von Javadoc und teilweise vom Compiler verschieden behandelt und ausgewertet
- Wichtige Tags sind
 - Allgemein: `@deprecated`, `@link`, `@see`
 - Bei Methoden: `@param`, `@return`

Javadoc für Methoden

```
/**  
 * Rechnet die gegebenen kmh in Meilen pro Stunde um,  
 * und gibt den berechneten wert zurück.  
 *  
 * @param kmh die umzurechnende Geschwindigkeit in km/h  
 *  
 * @return die Geschwindigkeit von kmh in Meilen pro Stunde  
 */
```

```
public static double berechneMph(double kmh) {  
    return kmh * 0.621;  
}
```

- *Immer* ein kurzer ganzer Satz, der beschreibt, **was** die Methode macht.
 - ◆ Erste Satz erscheint im Javadoc bei Methoden Übersicht
 - ◆ Gesamter Absatz erscheint dann bei Methoden Details
- Nie beschreiben, **wie** Methode implementiert ist; außer es ist für Benutzung relevant
- Für jeden Parameter *immer* ein `@param` Tag mit Name Parameter und nachfolgender *kurzen* Beschreibung des Parameters (Halbsatz reicht)
- Bei Methoden mit Rückgabewert *immer* ein `@return` Tag mit kurzen Beschreibung, was für ein Wert zurückgegeben wird (Halbsatz reicht)

Javadoc für Methoden

```
/**  
 * Rechnet die gegebenen kmh in Meilen pro Stunde um  
 * und gibt den berechneten Wert zurück.  
 * @param kmh die umzurechnende Geschwindigkeit in km/h  
 * @return die Geschwindigkeit von kmh in Meilen pro Stunde  
 *  
 * @deprecated aufgrund geänderter Namenskonventionen,  
 *             ab sofort immer {@link #getMph(double)} verwenden  
 */  
public static double berechneMph(double kmh) {  
    return kmh * 0.621;  
}
```

- Methoden, Klassen können im Lauf der Zeit veralten, da durch neue ersetzt
 - ◆ Diese mit `@deprecated` Tag kommentieren
 - ◆ Wird auch vom Compiler ausgewertet
- Verweise mit `@link` auf Klassen, Methoden (spärlich verwenden)
 - ◆ `{@link Geschwindigkeit}`: Hyperlink zur Javadoc der Klasse Geschwindigkeit
 - ◆ `{@link Geschwindigkeit#getMph(double)}`: Hyperlink zur Javadoc der Methode `getMph` in Klasse Geschwindigkeit

Method Summary

static double	berechneMph (double kmh) Deprecated. <i>aufgrund geänderter Namenskonventionen, ab sofort immer getMph(double) verwenden</i>
static double	getMph (double kmh) Gibt die Geschwindigkeit von kmh in Meilen Pro Stunde zurück.

- Bei veralteten Methoden erscheint Text von `@deprecated` in der Methodenübersicht, statt erster Satz des Javadoc Kommentars
- `@link` erscheint als Hyperlink zur Methode in Detailsicht
 - ◆ Bei Klassen erscheint ein Hyperlink zur Javadoc der Klasse

getMph

```
public static double getMph(double kmh)
```

Gibt die Geschwindigkeit von kmh in Meilen Pro Stunde zurück.

Parameters:

kmh - die umzurechnende Geschwindigkeit in km/h

Returns:

die Geschwindigkeit von kmh in Meilen pro Stunde