

Kapitel 1 – Datentypen: Zusammenfassung + Übungen

In diesem Kapitel lernst du die verschiedenen Datentypen in Java kennen, wie man Variablen deklariert und verwendet, und wie man mit einfachen Operationen arbeitet. Diese Grundlagen bilden die Basis für alles Weitere.

Themenübersicht

1. Variablen und Namenskonventionen

Variablen speichern Werte. Namen sollten sprechend sein (z. B. 'seitenAnzahl', nicht 'x').

2. Ganzzahlen – int

Der Typ int speichert ganze Zahlen, z. B. `int zahl = 10;`

3. Kommazahlen – double und float

double hat höhere Genauigkeit, float ist kleiner. Beispiel: `double d = 3.14;`

4. Zeichen – char

Ein einzelnes Zeichen, z. B. `char buchstabe = 'A';`

5. Wahrheitswerte – boolean

Kann true oder false speichern. Nützlich für Bedingungen: `boolean istVoll = false;`

6. Text – String

Speichert Text, z. B. `String name = "Anna";`

7. Arithmetische Operationen

`+`, `-`, `*`, `/`, `%` funktionieren mit Zahlen. Vorsicht: `5 / 2` ergibt 2, nicht 2.5 (Ganzzahldivision).

8. Modulo-Operator (%)

Gibt den Rest einer Division zurück. Beispiel: `5 % 2` ergibt 1.

9. Vergleichsoperatoren

`<`, `<=`, `==`, `!=`, `>=`, `>` für Bedingungen. Ergebnis ist ein boolean.

10. Logische Operatoren

`&&` (und), `||` (oder), `!` (nicht). Kombinieren mehrere Bedingungen.

11. Typecasts (Typumwandlung)

Explizit: `int x = (int) 3.5; // ergibt 3`

Implizit bei Erweiterung: `double d = 5;`

Übungsaufgaben

1. Deklariere eine Variable vom Typ `int` mit dem Namen `'alter'` und weise ihr den Wert 25 zu.
2. Erstelle eine `double`-Variable `'gewicht'` mit dem Wert 72.5 und gib sie mit `System.out.println()` aus.
3. Was ergibt `7 / 2` in Java? Und was ergibt `7.0 / 2`? Schreibe beide Ergebnisse aus.
4. Verwende `%` (Modulo), um zu prüfen, ob eine Zahl gerade ist.
5. Vergleiche zwei `int`-Werte und speichere das Ergebnis in einer `boolean`-Variable.
6. Baue einen Ausdruck mit `&&`, `||` und `!`, der `true` ergibt.
7. Führe einen expliziten Cast von `double` zu `int` durch und gib das Ergebnis aus.
8. Erstelle eine `String`-Variable und hänge mit `+` einen Nachnamen an (z. B. `"Max" + "Mustermann"`).