

Klassen mit unterschiedlichen Konstruktoren

Aufgabenstellung

1. Erstellen Sie Klassen mit sinnvollen Attributen und Methoden.
 2. Implementieren Sie Overloading von Konstruktoren (mehrere Konstruktoren mit unterschiedlicher Parameterliste).
 3. Erstellen Sie ein Klassendiagramm, z. B. in <https://app.diagrams.net/> , um die Struktur der Klassen zu visualisieren.
 4. Implementieren Sie die Klassen in Java.
-

Aufgabe 1: Buch

Attribute:

- `titel` (String)
- `autor` (String)
- `isbn` (String)
- `preis` (Double)

Methoden:

- `zeigeDetails()` – Gibt die Details des Buchs als String zurück.
- `istGuenstig()` – Gibt `true` zurück, wenn der Preis unter 20 liegt.

Konstruktoren:

1. **Ohne Parameter** – Initialisiert ein Standard-Buch.
 2. **Mit Titel und Autor.**
 3. **Mit allen Attributen.**
-

Aufgabe 2: Smartphone

Attribute:

- `marke` (String)
- `modell` (String)
- `speicher` (Integer in GB)
- `preis` (Double)

Methoden:

- `zeigeDetails()` – Gibt alle Details als String zurück.
- `preisNachRabatt(double rabattProzent)` – Berechnet und gibt den Preis nach einem Rabatt aus.

Konstruktoren:

1. **Ohne Parameter** – Standardwerte für Marke, Modell, Speicher und Preis.
 2. **Mit Marke und Modell.**
 3. **Mit allen Attributen.**
-

Aufgabe 3: Fahrrad

Attribute:

- `marke` (String)
- `typ` (String, z. B. „Mountainbike“ oder „Rennrad“)
- `ganganzahl` (Integer)
- `preis` (Double)

Methoden:

- `zeigeDetails()` – Gibt die Details des Fahrrads zurück.
- `hatVieleGaenge()` – Gibt `true` zurück, wenn die Ganganzahl größer als 10 ist.

Konstruktoren:

1. **Ohne Parameter** – Standardwerte für alle Attribute.
 2. **Mit Marke und Typ.**
 3. **Mit allen Attributen.**
-

Klassendiagramm

Erstellen Sie ein Diagramm mit den oben definierten Klassen in (<https://app.diagrams.net/>):

- Zeigen Sie Attribute, Methoden und Konstruktoren.
- Fügen Sie die Parameterlisten zu den Konstruktoren hinzu.