

## Aufgabe 1 – Zahlenreihe mit Bedingungen

Schreibe ein Programm, das alle Zahlen von **1 bis 50** ausgibt.

Ergänze folgende Logik:

- Wenn die Zahl durch 3 teilbar ist, gib zusätzlich „Foo“ aus.
  - Wenn die Zahl durch 5 teilbar ist, gib zusätzlich „Bar“ aus.
  - Wenn sie durch 3 **und** 5 teilbar ist, gib „FooBar“ aus.
- Nutze eine **for-Schleife** und **if-else-Kombinationen**

## Aufgabe 2

### Aufgabe 2.1 – Klasse Auto

Erstelle eine Klasse Auto mit den Attributen:

- marke (String)
- baujahr (int)
- geschwindigkeit (double)

und den Methoden:

- beschleunigen(double wert) → erhöht die Geschwindigkeit
- bremsen(double wert) → verringert die Geschwindigkeit
- toString() → gibt alle Werte schön formatiert aus

Verwende in den Methoden das Schlüsselwort **this**, um auf die Attribute des aktuellen Objekts zuzugreifen.

Teste die Klasse in einer separaten main-Klasse, indem du zwei Autos erstellst, sie unterschiedlich beschleunigst und ausgibst.

### Aufgabe 2.2 – Statische Attribute und Methoden

Erweitere die Klasse Auto:

- Füge ein **statisches Attribut** anzahlAutos hinzu, das zählt, wie viele Auto-Objekte erzeugt wurden.
- Erhöhe den Zähler im Konstruktor automatisch.
- Erstelle eine **statische Methode** getAnzahlAutos(), die den aktuellen Zählerwert zurückgibt.

### Aufgabe 2.3 – Klasse Elektroauto (Vererbung)

Erstelle eine Klasse Elektroauto, die **von Auto erbt**.

Füge hinzu:

- ein zusätzliches Attribut akkuStand (in %)
- eine Methode laden(int prozent), die den Akku um den angegebenen Wert erhöht (max. 100 %)
- Überschreibe die Methode toString(), sodass auch der Akkustand angezeigt wird.

Tipp: Verwende super.toString()

### Aufgabe 2.4 – equals() und Vergleich

Erstelle in der Klasse Auto eine eigene equals()-Methode:

- Zwei Autos gelten als **gleich**, wenn **Marke und Baujahr** übereinstimmen.
- Überschreibe equals(Object obj) korrekt.

Du kannst deine Implementierung mit der Testklasse aus Git testen. Die Testklasse erzeugt mehrere Autos und prüft mit equals(), ob sie gleich sind.

### Aufgabe 2.5 – toString() und Arrays

Erstelle ein Array aus mehreren Auto- und Elektroauto-Objekten.

Gib alle Objekte in einer Schleife aus und beobachte, welche toString()-Methode jeweils verwendet wird.