



## Aufgabe 1.1: Verwaltung von Autoteilen v2 (12 Punkte)

Überarbeiten Sie Ihr Programm aus Übung 7 und machen Sie es objektorientiert. Nach der Überarbeitung soll das Programm die gleiche Funktionalität haben, jedoch objektorientiert strukturiert sein.

Führen Sie folgende Änderungen durch:

- Speichern Sie die Arrays für Artikel und Aufträge als Objektfelder der Klasse *AutoPartWarehouse*.
- Führen Sie für alle Klassen Konstruktoren ein. Sie können direkt im Konstruktor der Klasse *Part* (bzw. *Order*) die erzeugten Objekte mit den Daten aus der Datei *parts.txt* (bzw. *orders.txt*) initialisieren.
- Machen Sie aus Klassenmethoden Objektmethoden, sodass z. B. aus dem Methodenaufruf *printOrder(order)* der Aufruf *order.print()* wird.

Testen Sie Ihr Programm mit folgender *main*-Methode:

```
public class AutoPartWarehouse {
    Part[] parts;
    Order[] orders;

    AutoPartWarehouse(String partsFile, String ordersFile) {
        parts = readParts(partsFile);
        orders = readOrders(ordersFile);
    }

    ...

    public static void main(String[] args) {
        AutoPartWarehouse warehouse =
            new AutoPartWarehouse("parts.txt", "orders.txt");
        warehouse.print();
    }
}
```

## Aufgabe 1.2: Bestellungen durchführen (12 Punkte)

Erweitern Sie das Programm aus Aufgabe 1.1 so, dass alle Aufträge, die erfüllbar sind, abgearbeitet werden.

Schreiben Sie dazu in der Klasse *Order* folgende Methoden:

- Die Methode *boolean isFulfillable()* prüft, ob der Auftrag mit dem aktuellen Lagerbestand durchführbar ist, d.h. ob von jedem Artikel eine ausreichende Stückzahl vorhanden ist.
- Die Methode *void process()* bucht die Artikel eines Auftrags aus dem Lager aus, d.h. die Lagerbestände werden entsprechend reduziert.

Testen Sie Ihr Programm mit folgender *main*- und *processOrders*-Methode:

```
public static void main(String[] args) {
    AutoPartWarehouse warehouse =
        new AutoPartWarehouse("parts.txt", "orders.txt");
    warehouse.print();
    warehouse.processOrders();
    Out.println("*** Warehouse after processing orders ***");
    Out.println();
    warehouse.print();
}

void processOrders() {
    for (int i = 0; i < orders.length; i++) {
        Order order = orders[i];
        if (order.isFulfillable()) {
            order.process();
            orders[i] = null;
        }
    }
}
```

Die Methode *processOrders* setzt erfüllte Aufträge im Auftrags-Array auf *null*. Passen Sie die Methode *printOrders* so an, dass *null*-Werte übersprungen werden.

Die Ausgabe soll wie folgt aussehen:

PARTS

| Part no. | Description                    | Stock |
|----------|--------------------------------|-------|
| 0        | VW Golf ABS Wheel Speed Sensor | 20    |
| 1        | Audi Q7 Wiper Motor Front      | 4     |
| 2        | Mercedes Fog Light Bulb        | 17    |
| 3        | Porsche 911 Front Oil Cooler   | 8     |

ORDERS

| Part no. | Description                    | Qty. |
|----------|--------------------------------|------|
| 0        | VW Golf ABS Wheel Speed Sensor | 6    |
| 1        | Audi Q7 Wiper Motor Front      | 3    |
| 2        | Mercedes Fog Light Bulb        | 2    |
| 3        | Porsche 911 Front Oil Cooler   | 3    |
| 1        | Audi Q7 Wiper Motor Front      | 9    |
| 2        | Mercedes Fog Light Bulb        | 2    |
| 3        | Porsche 911 Front Oil Cooler   | 5    |
| 1        | Audi Q7 Wiper Motor Front      | 1    |

\*\*\* Warehouse after processing orders \*\*\*

PARTS

| Part no. | Description                    | Stock |
|----------|--------------------------------|-------|
| 0        | VW Golf ABS Wheel Speed Sensor | 14    |
| 1        | Audi Q7 Wiper Motor Front      | 0     |
| 2        | Mercedes Fog Light Bulb        | 15    |
| 3        | Porsche 911 Front Oil Cooler   | 5     |

## ORDERS

| Part no. | Description                  | Qty. |
|----------|------------------------------|------|
| 1        | Audi Q7 Wiper Motor Front    | 9    |
| 2        | Mercedes Fog Light Bulb      | 2    |
| 3        | Porsche 911 Front Oil Cooler | 5    |

Geben Sie beide Aufgaben in einem einzigen Projekt ab. Testen Sie Ihr Programm zumindest mit den angegebenen Dateien *parts.txt* und *orders.txt*. JUnit-Tests sind in dieser Übung keine erforderlich.

## Aufgabe 1 - Verwaltung von Autoteilen

### Aufgabe 1/1 & 1/2: Java Program

Der Quellcode des Programms kann den beigefügten Dateien entnommen werden.

### Ausgabe

#### PARTS

| Part No. | Description                    | Stock |
|----------|--------------------------------|-------|
| 0        | VW Golf ABS Wheel Speed Sensor | 20    |
| 1        | Audi Q7 Wiper Motor Front      | 4     |
| 2        | Mercedes Fog Light Bulb        | 17    |
| 3        | Porsche 911 Front Oil Cooler   | 8     |

#### ORDERS

| Part No. | Description                    | Qty. |
|----------|--------------------------------|------|
| 0        | VW Golf ABS Wheel Speed Sensor | 6    |
| 1        | Audi Q7 Wiper Motor Front      | 3    |
| 2        | Mercedes Fog Light Bulb        | 2    |
| 3        | Porsche 911 Front Oil Cooler   | 3    |
| 1        | Audi Q7 Wiper Motor Front      | 9    |
| 2        | Mercedes Fog Light Bulb        | 2    |
| 3        | Porsche 911 Front Oil Cooler   | 5    |
| 1        | Audi Q7 Wiper Motor Front      | 1    |

\*\*\* Warehouse after processing orders \*\*\*

#### PARTS

| Part No. | Description                    | Stock |
|----------|--------------------------------|-------|
| 0        | VW Golf ABS Wheel Speed Sensor | 14    |
| 1        | Audi Q7 Wiper Motor Front      | 0     |
| 2        | Mercedes Fog Light Bulb        | 15    |
| 3        | Porsche 911 Front Oil Cooler   | 5     |

#### ORDERS

| Part No. | Description                  | Qty. |
|----------|------------------------------|------|
| 1        | Audi Q7 Wiper Motor Front    | 9    |
| 2        | Mercedes Fog Light Bulb      | 2    |
| 3        | Porsche 911 Front Oil Cooler | 5    |