



Aufgabe 1.1: Verwaltung von Autoteilen (12 Punkte)

Schreiben Sie ein Programm *AutoPartWarehouse*, mit dem man Autoteile verwalten kann. Ein Autoteil (Klasse *Part*) besteht aus einer Artikelnummer, einer Beschreibung und dem Lagerbestand in Stück:

```
class Part {
    int articleNr;
    String description;
    int stock;
}
```

Lesen Sie die Autoteile aus der Datei *parts.txt* ein und speichern Sie diese in einem Array. Die Datei hat folgenden Aufbau: Am Beginn steht die Anzahl der Autoteile, gefolgt von einer Zeile pro Teil mit der Artikelnummer, der Beschreibung und dem Lagerbestand:

```
4
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
```

Die Artikel sind dabei von 0 beginnend fortlaufend durchnummeriert, d.h. Sie können die Artikelnummer zum Indizieren im Array verwenden. Implementieren Sie eine Methode *readParts* zum Einlesen und eine Methode *printParts* zum Drucken der Autoteile mit folgenden Schnittstellen:

```
static Part[] readParts(String fileName)
static void printParts(Part[] parts)
```

Testen Sie Ihre Methoden mit folgender *main*-Methode:

```
public static void main(String[] args) {
    Part[] parts = readParts("parts.txt");
    printParts(parts);
}
```

Die Ausgabe von *printParts* soll wie folgt aussehen:

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

Aufgabe 1.2: Bestellungen für Autoteile (12 Punkte)

Erweitern Sie das Programm aus Aufgabe 1.1 und lesen Sie Bestellungen aus der Datei *orders.txt* ein. Eine Bestellung (Klasse *Order*) besteht aus mehreren Positionen und eine Position (Klasse *Item*) besteht aus einem Artikel (in der Datei steht die Artikelnummer, im Objekt speichern Sie eine Referenz auf den Artikel) sowie einer Bestellmenge:

```
class Order {
    Item[] items;
}
class Item {
    Part part;
    int quantity;
}
```

Speichern Sie Bestellungen in einem Array des Hauptprogramms. Eine Bestellung besteht nur aus einem Array von Positionen. Die Datei hat folgenden Aufbau: Am Beginn steht die Anzahl der Bestellungen, gefolgt von jeder Bestellung in einer eigenen Zeile. In jeder Zeile steht am Anfang die Anzahl der Positionen, gefolgt von Artikelnummer und Bestellmenge pro Position, durch Komma getrennt:

```
3
4 0,6 1,3 2,2 3,3
3 1,9 2,2 3,5
1 1,1
```

Implementieren Sie die Methoden *readOrders* zum Einlesen und *printOrders* zum Drucken der Bestellungen mit folgenden Schnittstellen. Speichern Sie zur Position eine Referenz auf den entsprechenden Artikel und übergeben Sie dazu das Artikel-Array an die Methode *readOrders*.

```
static Order[] readOrders(String fileName, Part[] parts)
static void printOrders(Order[] orders)
```

Testen Sie Ihre Methoden mit folgender *main*-Methode:

```
public static void main(String[] args) {
    Part[] parts = readParts("parts.txt");
    printParts(parts);
    Out.println();
    Order[] orders = readOrders("orders.txt", parts);
    printOrders(orders);
}
```

Die Ausgabe von *printOrders* soll wie folgt aussehen:

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

Hinweise zu Aufgabe 1.1 und 1.2:

- Globale, statische Variablen sind nicht erlaubt.
- Verwenden Sie die Klassen *Part*, *Order* und *Item* als reine Datenklassen (d.h. nur Felder, keine Methoden) und implementieren Sie alle Methoden in der Klasse *AutoPartWarehouse*.
- Sie können davon ausgehen, dass die Dateien *parts.txt* und *orders.txt* ausschließlich korrekte Daten enthalten.
- Die unter " " angegebene Beschreibung des Autoteils in der Datei *parts.txt* können Sie mit der Funktion *In.readString()* und das Komma in der Datei *orders.txt* mit der Funktion *In.read()* lesen.

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 nicht erforderlich.

Aufgabe 1 - Verwaltung von Autoteilen

Aufgabe 1/1: Java Program

Der Quellcode des Programms kann den beigefügten Dateien entnommen werden.

Aufgabe 1/2: Ausgabe

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

Aufgabe 2 - Bestellung für Autoteile

Aufgabe 2/1: Java Program

Der Quellcode des Programms kann den beigefügten Dateien entnommen werden.

Aufgabe 2/2: Ausgabe

Testweise wurde hier das Inputfile angepasst, um das Padding für die Quantity Spalte zu testen.

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	3657346
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