

Übungen – Klassenattribute und Strings

Aufgabe 1 – Erweiterung der Warenbestellung

Ausgangspunkt dieser Aufgabe ist der letzte Stand der Anwendung „**Warenbestellung**“. Es wird eine Klasse **Kunde** eingeführt und die Klasse Bestellung so erweitert, dass eine Bestellung für einen Kunden erstellt wird.

Außerdem werden die Kunden in zwei Gruppen eingeteilt. Die Kunden in der einen Gruppe bekommen einen Rabatt, die Kunden in der anderen Gruppe nicht. Dazu wird bei jedem Kunden ein Merkmal geführt, ob er auf die Warenpreise einen Rabatt bekommt oder nicht. Für alle Kunden, die Rabatt bekommen, gilt eine einheitliche Rabattregelung, definiert in der Klasse Kundenrabatt.

Die Klasse Kunde sieht so aus:

Kunde
- vorname : String - nachname : String - nummer : String - bekommtRabatt : boolean = false
<<Konstruktor>> +Kunde(vorname : String, nachname : String, nummer : String) <<Methoden>> +getVorname() : String +getNachname() : String +getNummer() : String +setBekommtRabatt(bekommtRabatt : boolean) +getBekommtRabatt() : boolean

Die Klasse Kundenrabatt:

Kundenrabatt
- <u>rabattsatz : double = 0.05</u>
+setRabattsatz(neuerRabattsatz : double) +getRabattsatz() : double +berechneRabattpreis(vollerPreis : double) : double

Beachten Sie, dass diese Klasse ausschließlich Klassenelemente hat.

Die Klassen-Attribute und Klassen-Methoden regeln das Rabattsystem. Alle Kunden bekommen den selben Rabatt, der in dieser Klasse festgelegt wird. Er kann mit den zugehörigen Klassen-Methoden gesetzt und abgefragt werden. Die Methode `berechneRabattpreis()` liefert den Rabattpreis auf der Basis des spezifizierten vollen Preises.

Die bestehende Klasse Bestellung erhält ein neues Attribut, das die Beziehung zu einem Kunde-Objekt darstellt. Die Beziehung wird im Konstruktor aufgebaut. Beides ist im folgenden Diagramm fett hervorgehoben:

Bestellung
- kunde : Kunde - positionen : Bestellposition[]
<<Konstruktor>> +Bestellung(kunde : Kunde) <<Methoden>> +nimmAuf(position : Bestellposition) +zeigeAn()

Außerdem muss in der Klasse Bestellung die Methode `zeigeAn()` angepasst werden.
 Beispiel für die Ausgabe eines Testlaufs, wobei die Zeilen, die mit `main()` beginnen, direkte Ausgaben aus dem Haupt-/Testprogramm sind:

```
main() - Zentral gesetzter Rabattsatz: 0.05

Bestellung von Dagobert Duck (KN 000351)
01019010 Hammer, Listenpreis: 19.0 EUR, Menge: 30
01019020 Zange, Listenpreis: 17.0 EUR, Menge: 20
01019030 Schraubendreher, Listenpreis: 12.0 EUR, Menge: 25

Bestellung von Emil Erpel (KN 000462) mit 5% Kundenrabatt.
03073073 Reifen, Listenpreis: 33.0 EUR, Kundenpreis: 31.35 EUR, Menge: 15
03073074 Schlauch, Listenpreis: 8.0 EUR, Kundenpreis: 7.6 EUR, Menge: 30
main() - Zentral gesetzter Rabattsatz: 0.05

Bestellung von Dagobert Duck (KN 000351)
01019010 Hammer, Listenpreis: 19.0 EUR, Menge: 30
01019020 Zange, Listenpreis: 17.0 EUR, Menge: 20
01019030 Schraubendreher, Listenpreis: 12.0 EUR, Menge: 25

Bestellung von Emil Erpel (KN 000462) mit 5% Kundenrabatt.
03073073 Reifen, Listenpreis: 33.0 EUR, Kundenpreis: 31.35 EUR, Menge: 15
03073074 Schlauch, Listenpreis: 8.0 EUR, Kundenpreis: 7.6 EUR, Menge: 30
03073103 Luftpumpe, Listenpreis: 13.0 EUR, Kundenpreis: 12.35 EUR, Menge: 5

main() - Es wird ein neuer Rabattsatz gesetzt...
main() - Zentral gesetzter Rabattsatz: 0.07

Bestellung von Daisy Duck (KN 000573) mit 7% Kundenrabatt.
02042051 Rock, Listenpreis: 133.0 EUR, Kundenpreis: 123.69 EUR, Menge: 5
02042061 Bluse, Listenpreis: 83.0 EUR, Kundenpreis: 77.19 EUR, Menge: 4
02042071 Bustier, Listenpreis: 112.0 EUR, Kundenpreis: 104.16 EUR, Menge: 3
```

Aufgabe 2 – Anpassung der Ausgabe

Sorgen Sie mit den neuen Möglichkeiten für formatierte Ausgabe dafür, dass die Listen sauber und gut lesbar ausgegeben werden. Beispielsweise so:

```
Bestellung von Dagobert Duck (KN 000001)
  Nummer    Bezeichnung    Listenpreis    Menge
01019010    Hammer          19,00         30
01019020    Zange             17,00         20
01019030    Schraubendreher    12,00         25

Bestellung von Emil Erpel (KN 000002) mit 5,0% Kundenrabatt.
  Nummer    Bezeichnung    Listenpreis    Kundenpreis    Menge
03073073    Reifen          33,00         31,35         15
03073074    Schlauch         8,00          7,60          30
03073103    Luftpumpe        13,00         12,35          5
```

Aufgabe 3 – Format überprüfen

Die Warennummern sollen im System geändert werden. So etwas passiert sehr häufig und sorgt in der IT-Abteilung immer für Ärger. Wir behelfen uns diesmal damit, neben der `getWarennummer()`-Methode eine zweite Methode einzuführen: `getNormalisierteWarennummer()`. Diese soll intern überprüfen, ob die Ware schon im neuen 12-Stelligen Format „DE-0000-0000“ vorliegt. Wenn ja, dann soll einfach die Ausgabe erfolgen, wenn nein, dann zerlegen Sie die bisherige Warennummer mit Hilfe eines regulären Ausdrucks und setzen die zerlegten Elemente so zusammen, dass die Ausgabe nun das neue Format hat. Der Umbau soll nur für die Ausgabe erfolgen und (noch) nicht den Datensatz ändern. Beispiel:

```
Bestellung von Dagobert Duck (KN 000001)
  Nummer    Bezeichnung    Listenpreis    Menge
01019010    Hammer          19,00         30
01019020    Zange             17,00         20
01019030    Schraubendreher    12,00         25
```

wird zu

```
Bestellung von Dagobert Duck (KN 000001)
  Nummer    Bezeichnung    Listenpreis    Menge
DE-0101-9010    Hammer          19,00         30
DE-0101-9020    Zange             17,00         20
DE-0101-9030    Schraubendreher    12,00         25
```

Aufgabe 4 – Die Klasse Bruchzahl leicht modifiziert

Ausgangspunkt dieser Aufgabe ist die Klasse `Bruchzahl` aus den Übungen zur letzten Vorlesung.

Die Methode `ggT()` operiert nicht auf einem aktuellen Objekt, sondern nur auf Argumenten. Man kann sie also als Klassen-Methode implementieren. Da sie nicht nur für die Klasse `Bruch` von Interesse ist, lagern wir sie als Klassen-Methode in eine separate Klasse aus, die wir `Algorithmen` nennen. Die Klasse `Bruchzahl` muss entsprechend angepasst werden.

Spezialaufgabe – Code-Qualität

Aufgabe 5 – Einfache Änderungen

Suchen Sie sich einen Programmierpartner, mit dem Sie Quellcode austauschen. Ausgangspunkt sind die Resultate der Aufgabe 3 in dieser Übung. Idealerweise hat der Partner nicht mit Ihnen zusammen die Übungen durchgeführt und noch keinen Einblick in Ihren Code gehabt.

Geben Sie den Sourcecode für das Resultat in Übung 3 an Ihren Partner und lassen Sie sich von ihr/ihm ebenfalls das Resultat geben.

Ab jetzt gilt: stoppen Sie die Zeiten, wenn Sie die nachfolgenden Aufgäbchen durchführen.

Schritt 1

Ändern Sie die Reihenfolge der Liste, so dass die Menge in der Spalte links vom Listenpreis steht.

Schritt 2

Geben Sie in einer Spalte ganz rechts den Gesamtpreis für die Position aus (30x Hammer à 19,00 EUR macht 570,00 EUR)

Schritt 3

Berechnen Sie den Gesamtpreis (Netto) für alle Positionen und geben Sie diesen unterhalb der Liste an. Weiterhin weisen Sie 19% MwSt. aus und berechnen Sie den insgesamten (Brutto) Preis und zeigen Sie zusätzlich beides untereinander an.

Aufgabe 6 – Resümée 1

Dokumentieren Sie sich separat die 3 Zeiten für die Änderungen.

Wenn Sie jetzt im nachhinein darüber sinnieren, ob etwas und wenn ja, was genau es Ihnen erschwert hat, die Änderungen durchzuführen, kommen Sie ggf. zu gewissen Schlüssen, was auf den Leser von fremden Code eher unglücklich oder eher erleichternd wirkt.

Machen Sie sich Notizen, wir sprechen die DOs und DON'Ts in der nächsten Vorlesung an.

Aufgabe 7 – Back to the Roots und Resümée 2

Holen Sie sich Ihren inzwischen modifizierten Quellcode wieder zurück und geben Sie Ihrem Partner das Resultat Ihrer Bemühungen.

Hat sich für Sie viel verändert? Kommen Sie nun besser oder schlechter mit dem Code zurecht? Machen Sie sich auch hier Notizen.

Um den Lerneffekt aus Aufgabe 6 und 7 nicht gleich verfliegen zu lassen, ändern wir gleich nochmal etwas, diesmal aber an unserem eigenen Code...

Schritt 4

Veränderung in der Klasse Bestellung. Ändern Sie die Bestellung so ab, dass die Lösung mit der Aufbewahrung in einem Array nicht mehr limitiert ist, d.h. bei Bedarf muss das bisherige Array (besser: der Inhalt) in ein neues, größeres Array umkopiert werden.

Es gibt mehrere Strategien, aber bitte verwenden Sie ausschließlich Arrays und keine anderen Hilfsmittel aus der Java-Plattform.

Notieren Sie sich, ob Sie bei der Umsetzung jetzt mit Hinblick auf die gemachten Erfahrungen etwas anders gemacht haben.