Prof. Dr. Stefan Brass
Dipl.-Inform. Annett Thüring
Dipl.-Inform. Steffen Schiele
Dipl.-Inform. Andreas Bienert

Martin-Luther-Universität Halle-Wittenberg
Institut für Informatik

Wintersemester 2014/15

Objektorientierte Programmierung: Hausaufgabenblatt 8

Abgabe: 18.12.2014, 12:00 — Übung: 12./13.01.2015

Hausaufgabe 8: (4 Theoriepunkte)

Schreiben Sie eine Klasse zur Verwaltung von Angeboten für Artikel (die Beispiel-Daten sind Feuerwerksartikel, genauer Feuerwerks-Batterien). Die für diese Aufgabe wesentlichen Daten eines Artikels sind:

- Name des Herstellers (oder Importeurs), z.B. Weco, Jorge, Blackboxx.
- Name des Artikels, z.B. “Tanz der Vampire”.
- Minimaler Preis, zu dem der Artikel angeboten wird.
- Maximaler Preis, zu dem der Artikel angeboten wird.
- Name des Webshops oder Ladens, der den Artikel zum minimalen Preis anbietet.

Ihre Klasse soll daher folgende Schnittstelle anbieten (diese Komponenten sollen `public` sein, alles andere `private`):

- Ein Konstruktor mit vier Parametern: `hersteller`, `artikelname`, `preis`, `anbieter`. Beim Konstruktor-Aufruf wird also nur ein Preis übergeben, der zunächst gleichzeitig minimaler und maximaler Preis ist. Mit der folgenden Methode `alternativAngebot` werden anschließend weitere Angebote erfasst. Preise werden als ganze Zahl (Cent-Wert) übergeben. Da es beim Datentyp `double` etwas unübersichtliche Ungenauigkeiten gibt, ist das so besser.
- Eine Methode `alternativAngebot` mit den beiden Parametern `preis` (ganze Zahl) und `anbieter` (Zeichenkette). Diese Methode liefert keinen Wert zurück, sondern speichert nur die Daten im Objekt: Falls der Preis kleiner ist als der bisherige Minimalpreis, muss der Minimalpreis aktualisiert werden. Dann soll auch der bisher billigste Anbieter ersetzt werden. Entsprechend muss auch der Maximalpreis verändert werden, wenn der neue Preis oberhalb der bisherigen Preisspanne liegt. Ist der Preis gleich dem bisherigen Minimalpreis, brauchen Sie den im Objekt gespeicherten günstigsten Anbieter nicht zu ändern.
- Eine Methode `hersteller` ohne Parameter, die den Hersteller liefert.
- Eine Methode `artikelname` ohne Parameter, die den Artikelnamen liefert.
- Eine Methode `minPreis` ohne Parameter, die den minimalen Preis liefert (als ganze Zahl in Cent).

- Eine Methode `maxPreis` ohne Parameter, die den maximalen Preis liefert (auch in Cent).
- Die Methode `toString` ohne Parameter, die die wesentlichen Daten des Objektes als String liefert. Halten Sie sich dabei an das folgende Format:

  ```
  Hersteller Artikelname: Preisspanne (Anbieter).
  ```

  Falls minimaler und maximaler Preis übereinstimmen, soll als `Preisspanne` nur dieser eine Preis ausgegeben werden, natürlich in Euro mit zwei Nachkomma-Stellen. Falls es eine echte Preisspanne gibt, tragen Sie Minimalpreis, “ - ”, Maximalpreis in den obigen String ein. Ein Beispiel für ein Ergebnis von `toString()` ist:

  ```
  Weco Tanz der Vampire: 9.95 - 11.99 (Roeder)
  ```

  Beachten Sie, dass die Methode `toString()` den Text als Zeichenkette zurückliefern soll, und nicht selbst ausdrucken. Noch ein Tipp: Sie können einen Cent-Betrag in das gewünschte Format umwandeln mit:

  ```
  String s = String.format("%d.%02d", preis/100, preis%100);
  ```

Fügen Sie Ihre Klassen oben in das folgende Rahmenprogramm ein:

`http://www.informatik.uni-halle.de/~brass/oop14/homework/Artikel.java`

Sie müssen sich außerdem folgende Daten-Datei kopieren:

`http://www.informatik.uni-halle.de/~brass/oop14/homework/Artikel.txt`

Jede Zeile der Daten-Datei entspricht einem Angebot. Z.B. gibt es folgenden Datensatz:

```
Weco|Tanz der Vampire|9.95|Roeder
```

Das Rahmenprogramm liest die Datei ein, zerlegt jede Zeile in die vier Bestandteile, und prüft dann, ob es schon einen Artikel mit dem Hersteller und Namen gibt. Falls ja, wird ein alternatives Angebot gespeichert. Falls nein, wird ein neues Objekt Ihrer Klasse erzeugt und in einem Array gespeichert. Es empfiehlt sich natürlich, sich auch das Rahmenprogramm durchzulesen. Einerseits könnten Sie so z.B. die Arbeit mit Dateien wiederholen, andererseits finden sich dort ja Aufrufe der Methoden, die Sie entwickeln sollen.

Noch ein Hinweis: In der Datei `Artikel.java` steht nicht nur Ihre Klasse `Artikel`, sondern auch die Klasse `ArtikelTest` (das Rahmenprogramm). Der Java-Compiler erzeugt pro Klasse eine `.class`-Datei. Nur `ArtikelTest` hat eine `main`-Methode. Sie müssen das Programm also mit folgendem Befehl ausführen:

```
java ArtikelTest
```

(Die Preise in der Datei waren die Angabe im jeweiligen Webshop für ausgewählte Artikel. Allerdings kommt für den Gefahrgut-Transport noch ein relativ kräftiger Aufschlag hinzu, so dass der billigste Einzelpreis nicht unbedingt den günstigsten Gesamtpreis garantiert. Manche haben auch noch hier nicht eingerechnete Rabatte. Bei vielen Feuerwerksartikel-Webshops kann man schon keine Bestellungen mehr für Silvester aufgeben.)

Übungsaufgabe 8A: (ohne Abgabe)

Bitte bearbeiten Sie die Übungsaufgaben auf dem Hausaufgabenblatt, aber geben Sie diese Aufgaben nicht ab. Diese Aufgaben werden in der Übung besprochen. Sie müssen Ihre Lösung eventuell in der Übung vorführen.

Das folgende Programm enthält mindestens 5 Fehler. Finden Sie möglichst viele davon.

```
class C {
        private int n;
        private static int i = 10;

        public C(int i) {
                System.out.println(n);
                this.n = i;
        }

        public int m() {
                return this.n * i;
        }

        public static int s() {
                return m() + C.i;
        }
}

class Test {
        public static void main(String[] args) {
                C obj = new C();
                C x = null;
                Test y = new Test();
                System.out.println(obj.n);
                int n = C.s() + obj.m();
                System.out.println(n);
                System.out.println(x.m());
                System.out.println(y.m());
        }
}
```


Übungsaufgabe 8B: (ohne Abgabe)

Angenommen, Sie wollen eine Klasse für ganze Zahlen beliebiger Länge entwickeln (`int`-Werte sind ja in der Größe begrenzt). Konkret wollen Sie diese Klasse benutzen, um die Fakultäts-Funktion ($n! = 1 * 2 * 3 * \cdots * n$) auch für größere n berechnen zu können. Die Fakultäts-Funktion wächst ja sehr schnell.

Welche Konstruktoren und Methoden sollte Ihre Klasse anbieten, so dass es mindestens für die Fakultäts-Funktion reicht? Sie sollen die Klasse selbst nicht programmieren, allenfalls die Köpfe der Methoden angeben (also die Schnittstelle der Klasse beschreiben).

Auch diese Aufgabe bitte nicht einsenden, aber bis zur Übung bearbeiten.