Ludwig-Maximilians-Universität München
Institut für Informatik
PD. Dr. Peer Kröger, Dr. Andreas Züfle
Daniel Kaltenthaler, Klaus Schmid

München, 23.12.2014

Einführung in die Programmierung

WS 2014/15

Übungsblatt 11: Objektorientierte Programmierung

Besprechung: 14./16./19.01.2015

Ende der Abgabefrist: Dienstag, 13.01.2015 14:00 Uhr.

Hinweise zur Abgabe:
Geben Sie bitte Ihre gesammelten Lösungen zu diesem Übungsblatt in einer Datei `loesung11.zip` unter `https://uniworx.ifi.lmu.de` ab.
Alle Code-Aufgaben müssen als .java abgegeben werden. PDF-code wird nicht akzeptiert. Nicht kompilierbare Abgaben werden mit 0 Punkten bewertet.

Die in diesem Blatt vergebenen Punkte sind Bonus-Bonuspunkte: sie gehen nicht in die erreichbare Gesamtpunktzahl mit ein, sondern erhöhen Ihre erreichten Übungspunkte direkt.

Aufgabe 11-1 *Umrechnung von Zeiteinheiten* **2+2 Punkte**

(a) Implementieren Sie in einer Datei `Umrechnung.java` eine statische Methode `int[] umrechnen(int sekunden)`, die eine Anzahl (> 0) von Sekunden als Parameter übergeben bekommt, diese in Stunden, Minuten und Sekunden umrechnet und diese Werte als `int`-Array der Länge 3 zurückgibt. Der Aufruf von `umrechnen(64)` gibt also `{0,1,4}`, der Aufruf von `umrechnen(3600)` gibt `{1,0,0}` zurück.

Schreiben Sie in der selben Klasse eine `main`-Methode, so dass zum Testen der Methode `umrechnen` beim Aufruf der Klasse `Umrechnung` über die Kommandozeile eine Anzahl von Sekunden eingegeben werden kann. Die Ausgabe der `main`-Methode auf die Kommandozeile soll nach folgendem Schema erfolgen:
`XXX Sekunden ergeben X Stunden, X Minuten und X Sekunden.`
Hierbei ist auf eine korrekte und sinnvolle Ausgabe bzgl. der Verwendung von Einzahl/Mehrzahl und Vorkommen der verschiedenen Zeiteinheiten zu achten.

Beispiel: Der Aufruf
`java Umrechnung 64`
gibt
`64 Sekunden ergeben 1 Minute und 4 Sekunden.`
auf die Kommandozeile aus.

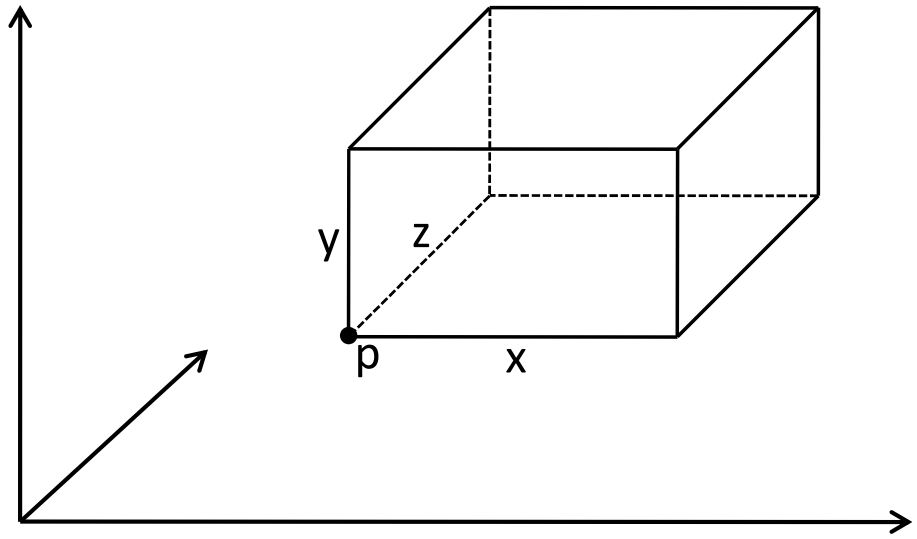
(b) Die Methode `umrechnen` gibt die errechneten Werte von Stunden, Minuten und Sekunden als `int`-Array der Länge 3 aus. Wie könnte man dies (sinnvoller) objektorientiert modellieren?

Implementieren Sie Ihre Lösung in einer Klasse `Umrechnung2`.

Geben Sie für diese Aufgabe ein jar-Archiv `11-1.jar` ab, das die Dateien `Umrechnung.java`, `Umrechnung2.java` und gegebenenfalls weitere Klassen für Teilaufgabe (b) enthält.

Aufgabe 11-2 *Objektorientierte Modellierung* **3+3 Punkte**

Bei Quadern handelt es sich um Objekte, deren Kanten achsenparallel in einem dreidimensionalen Koordinatensystem liegen. Quader werden spezifiziert (siehe Bild) durch einen der Eckpunkte `p` und die Länge der an `p` angrenzenden Kanten `x` (parallel zur x-Achse), `y` (parallel zur y-Achse) und `z` (parallel zur z-Achse).



Hinweis: Achten Sie im Folgenden in der gesamten Aufgabe auf sinnvolle Datenkapselung.

(a) Implementieren Sie eine Klasse `Punkt3D` zur Verwaltung dreidimensionaler Punkte in einem reellen Koordinatensystem als Erweiterung der Klasse `Punkt2D`, die Sie auf der Vorlesungs-Homepage finden:

Definieren Sie neben einem geeigneten Konstruktor und geeigneten Attributen die folgende Methode:
`void verschiebe(double x, double y, double z)`,
die den Punkt um `x` entlang der x-Achse, um `y` entlang der y-Achse und um `z` entlang der z-Achse verschiebt.

(b) Implementieren Sie eine Klasse `Quader`, die die Klasse `Punkt3D` aus der vorherigen Teilaufgabe sinnvoll verwendet. Die Klasse soll einen geeigneten Konstruktor und geeignete Attribute bereitstellen. Implementieren Sie für die Klasse Quader zusätzlich folgende Methode:
`void verschiebe(double x, double y, double z)`,
die den Quader um `x` entlang der x-Achse, um `y` entlang der y-Achse und um `z` entlang der z-Achse verschiebt.

Aufgabe 11-3 *Bruchrechnen mit Java* **0 Punkte**

In dieser Aufgabe soll eine Klasse `Bruch` implementiert werden, die Brüche (rationale Zahlen) und die Grundrechenarten für Brüche implementiert.

(a) Definieren Sie eine Klasse `Bruch`, die (gewöhnliche) Brüche mit Zähler und Nenner vom Typ **`int`** repräsentiert. Deklarieren Sie dazu entsprechende Attribute.

(b) Erweitern Sie die Klasse `Bruch` um einen Konstruktor `Bruch(`**`int`**` n)`, der eine ganze Zahl `n` als Bruch erzeugt.

(c) Erweitern Sie die Klasse `Bruch` um eine Methode `String toString()`, die ein `String`-Objekt zurückgibt, das einen entsprechenden Bruch geeignet textuell repräsentiert.

(d) Erweitern Sie die Klasse `Bruch` um eine Methode `Bruch negiere()`, die einen neuen Bruch erzeugt, so dass gilt: Ist b ein Bruch mit Wert z/n, so ist `b.negiere()` ein Bruch mit Wert $-z/n$.

(e) Erweitern Sie die Klasse `Bruch` um folgende Methoden:

- `Bruch addiere(Bruch b)`
- `Bruch subtrahiere(Bruch b)`
- `Bruch multipliziere(Bruch b)`
- `Bruch dividiere(Bruch b)`

Jede dieser Methoden soll einen neuen Bruch zurückgeben, der den Wert der Berechnung repräsentiert. Es gilt:

$$\frac{z_1}{n_1} + \frac{z_2}{n_2} = \frac{z_1 n_2 + z_2 n_1}{n_1 n_2}, \frac{z_1}{n_1} - \frac{z_2}{n_2} = \frac{z_1 n_2 - z_2 n_1}{n_1 n_2}, \frac{z_1}{n_1} \times \frac{z_2}{n_2} = \frac{z_1 z_2}{n_1 n_2}, \frac{z_1}{n_1} / \frac{z_2}{n_2} = \frac{z_1 n_2}{n_1 z_2}$$

Testen Sie diese Methoden für geeignete Eingaben.

(f) Erweitern Sie die Klasse `Bruch` um eine Methode **`boolean`** `kleinerAls(Bruch b)`, die folgende Werte zurückgibt: **`true`**, wenn `b` grösser ist als der Wert des aufrufenden Objekts vom Typ `Bruch`, **`false`** sonst.

(g) Erweitern Sie die Klasse `Bruch` um eine Methode **`double`** `alsDouble()`, die das aufrufende Objekt vom Typ `Bruch` in eine Gleitkommazahl umwandelt.

Aufgabe 11-4 *Der Grinch hasst die Winterzeit* **1+3+4+2 Punkte**

Der Grinch hasst die Winterzeit. Überall feiern die Familien ihre Feste, sei es Weihnachten, Chanukka, Kwanzaa oder einfach so mit der Familie. Um seinem Ärger Luft zu machen, möchte er möglichst vielen Menschen vor dem Jahreswechsel die Laune verderben. In den auf der Homepage verfügbaren Klassen `Grinch.java`, `Haus.java` und `Geschenk.java` hat er sich einen Plan ausgedacht, und nun ist es Ihre Aufgabe, ihm zu helfen.

(a) Es ist der 23. Dezember und der Grinch möchte loslegen. Er weiß, dass er nur bis zum 31. Dezember Zeit hat, denn da ist er zu einer Neujahrsparty auf Gran Canaria eingeladen. Der Grinch arbeitet nur 10 Stunden jede Nacht.
Erweitern Sie die Klasse `Grinch` um eine Variable `restzeit`, die die verbleibenden Stunden als Ganzzahl hält. Für jeden erzeugten Grinch soll die Restzeit mit 80 Stunden initialisiert werden. Es wird nicht zwischen einzelnen Nächten unterschieden.

(b) In jedem Haus sind Geschenke verteilt. Manche liegen unter dem Baum, manche im Schrank, manche auf dem Tisch. Der Grinch möchte sie alle mitnehmen.
Erweitern Sie die Klasse `Haus` um eine Methode `ausraeumen()`, die alle Geschenke aus dem Haus entfernt (aus den Arrays `baum`, `schrank` und `tisch`) und gesammelt in einem neuen Array zurück gibt. Der Zugriff sollte auf folgende Weise möglich sein:

```
Haus h = new Haus();
Geschenk[] geschenke = h.ausraeumen();
```

(c) Geschenke klauen kostet Zeit. Für alle angefangenen zehn Geschenke pro Haus veranschlagt der Grinch eine Stunde. Wenn also das `ausraeumen()` eines Hauses 54 Geschenke ergibt, hat der Grinch 6 Stunden in diesem Haus verbracht. Die Häuser der Stadt hat sich der Grinch alle in seiner Klassenvariable `stadt` gemerkt.
Erweitern Sie die Methode `aufTourGehen()` in der Klasse `Grinch` so, dass der Grinch nach und nach die Häuser der Stadt abläuft und jedes Haus ausräumt. Die erbeuteten Geschenke sichert sich der Grinch in seiner Variable `geklauteGeschenke`. Eine Methode, zwei Arrays zu kombinieren, haben Sie in der Vorlesung schon kennengelernt.
Achten Sie darauf, dass der Grinch sein Zeitkonto nicht überschreitet. Mit jedem besuchten Haus soll das Zeitkonto um die passende Anzahl Stunden schrumpfen. Ist die Zeit abgelaufen, bricht der Grinch seinen Diebeszug ab. Die Geschenke des zuletzt besuchten Hauses, während dem die Zeitüberschreitung passiert, sollen dabei verworfen werden.

(d) Zurück in seinem Versteck, will der Grinch seine geklauten Geschenke zählen.
Implementieren Sie in der Klasse `Grinch` eine neue Methode `inventur()`, die die Anzahl und das Gesamtgewicht der erbeuteten Geschenke auf der Standardausgabe ausdruckt.