SCHOOL-SCOUT.DE

Unterrichtsmaterialien in digitaler und in gedruckter Form

Auszug aus:

Einführung in Java - Lehrerband

Das komplette Material finden Sie hier:

School-Scout.de

School-Scout — Der persönliche Schulservice

Einführung in Java

Arbeitsbuch für Schülerinnen und Schüler der gymnasialen Oberstufe

Lehrerband

Dr. Daniel W. Appel
Cecilien-Gymnasium Düsseldorf

Düsseldorf, März 2013

Vorwort

Dieses Arbeitsbuch bietet einen Einstieg in die Programmiersprache Java für Schülerinnen und Schüler der gymnasialen Oberstufe.

Schritt für Schritt werden anhand zahlreicher Übungen und drei unterschiedlich umfangreicher Programmierprojekte die wichtigsten imperativen Grundkonzepte eingeführt. Von Beginn an wird Wert darauf gelegt, diese Übungen möglichst alltagsnah zu gestalten und alle Konzepte anschaulich darzustellen.

Besondere Vorkenntnisse sind keine nötig. Die Unterrichtsgruppen, für die die vorliegenden Materialien ursprünglich erstellt wurden, hatten zuvor im Differenzierungsbereich der Stufen 8 und 9 zwei Jahre Informatikunterricht. Darauf wird hier aber nicht explizit Bezug genommen.

An der Schule des Autors wird Eclipse als Programmierumgebung eingesetzt. Auf eine Einführung darin wird aber bewusst verzichtet, da viele Schulen andere Werkzeuge bevorzugen. Einen expliziten Bezug darauf gibt es daher nur sehr am Rande, so dass dieses Arbeitsbuch ohne Probleme auch mit anderen Programmierumgebungen (oder sogar einem einfachen Texteditor) eingesetzt werden kann.

Es wird bewusst auf Konzepte des objektorientierten Programmierens (OOP) verzichtet. Dies hat verschiedene Gründe. Ein erster Grund ist, dass ein sicherer Umgang mit Grundlagen der Programmiersprache Java — wie er hier vermittelt wird — einen späteren Einstieg in die OOP wesentlich erleichtern kann, da man sich so tatsächlich auf die Konzepte der Objektorientierung konzentrieren kann und nicht über andere Hindernisse (wie etwa ein vergessenes Semikolon oder andere syntaktische Schwierigkeiten) stolpert. Weiterhin kann man auf diese Weise, die Klassen, die man später modelliert und implementiert, sehr viel schneller mit interessantem Leben füllen.

Ein weiteres Arbeitsbuch, das basierend auf dem hier vorliegenden einen Einstieg in die OOP bietet, ist derzeit in Arbeit.

Alle verwendeten Graphiken wurden selbst erstellt. Bei der Erstellung der Flussdiagramme wurde mit freundlicher Genehmigung der Verfassers, Herrn F. Folkmann, das kostenlose Werkzeug *PapDesigner* verwendet, das man unter `http://www.friedrich-folkmann.de` erhalten kann.

Alle Materialien wurden bereits im Unterricht eingesetzt und aufgrund praktischer Erfahrungen ergänzt und ausgebessert. Dies ist natürlich ein andauernder Prozess, so dass auch in Zukunft Aktualisierungen dieses Arbeitsbuches zu erwarten sind.

Anregungen und Verbesserungsvorschläge sind herzlich unter `appel@ceci.de` willkommen.

Dr. D. Appel Düsseldorf, den 10. März 2013

Inhaltsverzeichnis

1 Java als Taschenrechner

1.1 Ganze Zahlen

Aufgabe 1. Hier berechnet Java zunächst die Summe $5 + 13 = 18$ und gibt diese anschließend aus.

Aufgabe 2. Hier sollten die Schüler sich anhand verschiedener Beispiele, wie etwa

```
System.out.println(15-3);
System.out.println(1-24);
System.out.println(20*2);
System.out.println(0*3);
```

klarmachen, wie man Subtraktion und Multiplikation ganzer Zahlen in Java erreichen kann.

Aufgabe 3. Sie wird 3 Stücke erhalten (denn die 2 passt dreimal in die 7). Es bleibt 1 m Abfall übrig (denn $7 - 3 \cdot 2 = 1$).

Aufgabe 4. (a) Diese Anweisungen liefern die Werte 5, 3, 10 und 0 (da die Divisionen jeweils ohne Rest aufgehen).

(b) Hier erhält man natürlich eine Fehlermeldung, da man nicht durch Null dividieren darf. Genauer erhält man die Meldung `Exception in thread \main\ java.lang.ArithmeticException: / by zero` gefolgt man einer genauen Angabe, wo dieser Fehler auftritt.

(c) Man erhält die Ergebnisse 6, 4, 5 und 3. Hier wird eine ganzzahlige Division durchgeführt, d.h. Nachkommastellen werden abgeschnitten. Die letzte Rechnung sollte dabei an Aufgabe 3 erinnern.

(d) Hier erhält man die Ergebnisse 1, 1, 2 und 1. Es wird jeweils genau der Rest angegeben, der bei einer ganzzahligen Division entsteht. Die letzte Rechnung sollte dabei wieder an Aufgabe 3 erinnern.

Aufgabe 5. Die Anzahl der Stücke (es sind 56) berechnet man mit dieser Anweisung:

```
System.out.println(734/13);
```

Die Menge des Abfalls (der 6 cm beträgt) entsprechend mit dieser:

```
System.out.println(734%13);
```

Aufgabe 6. Mit

```
System.out.println(143%24);
```

berechnet man, dass der Großrechner um 23 Uhr fertig war. Mit

```
System.out.println(143/24);
```

stellt man fest, dass er fünf volle Tage (und 23 Stunden) beschäftigt war.

Aufgabe 7. Die Rechnung

```
System.out.println((14+17)%24);
```

zeigt, dass es 7 Uhr morgens sein wird.

Aufgabe 8. Die Anweisung

```
System.out.println(<auszugebene Daten>);
```

führt dazu, dass neue Ausgaben in einer neuen Zeile beginnen werden. Bei der Anweisung

```
System.out.print(<auszugebene Daten>);
```

hingegen findet kein Zeilenumbruch statt.

Aufgabe 9. Hier sollten die Schüler weitere Beispiele finden. Zum Beispiel Angabe eines Datums oder des Wochentages in einer bestimmten Anzahl von Tagen. Die Wochentage kann man dazu einfach von 0 bis 6 nummerieren.

1.2 Dezimalzahlen

Aufgabe 10. Wie erwartet erhält man das Ergebnis $7,6$.

Aufgabe 11. Hier erhält man das Ergebnis $3,4$ und auch weitere Beispiele sollten deutlich manchen, dass dies die gewöhnliche Division ist. Allerdings kann es in einigen Fällen zu Rundungsfehlern kommen.

Aufgabe 12. Sie wird 2 Stücke erhalten und es bleiben $0,1$ m Abfall übrig.

Aufgabe 13. Die Anweisung

```
System.out.println(2.5 % 1.2);
```

liefert den Rest, der übrig bleibt. (Auch hier kann ein Rundungsfehler auftreten. Inwieweit man dies problematisiert, sollte man aufgrund des Interesses und der Leistungsstärke des Kurses entscheiden.)

Aufgabe 14. Wir verwenden die beiden Anweisungen

```
System.out.println(734.5 / 13.2);
System.out.println(734.5 % 13.2);
```

Erstere liefert den Wert 55.643939..., die zweite den Wert $8,5$. D.h. sie wird 55 Stücke erhalten und es bleiben $8,5$ m Abfall übrig.

Aufgabe 15. Die Anweisung

```
System.out.println((14.75 +17.5)%24);
```

liefert den Wert $8,25$, d.h. es wird 8:15 Uhr sein.

Aufgabe 16. Dies erreicht man zum Beispiel mit jeder der folgenden Anweisungen:

```
System.out.println(22.0 / 5);
System.out.println(22. / 5);
System.out.println(22 / 5.0);
System.out.println(22 / 5.);
```

Aufgrund des Punktes verwendet Java hier Fließkommazahlen.

2 Variablen in Java

2.1 Das Konzept einer Variablen

Aufgabe 1. (a) Dies erreicht man mit diesen Anweisungen:

```
int meineZahl;
meineZahl = 5;

int meineZweiteZahl;
meineZweiteZahl = 10;
```

(b) Man erhält die Werte 15, −5, 50, 0, 5. Dies sind die Ergebnisse der fünf Rechnungen, wobei jedes mal die gespeicherten Werte der Variablen verwendet werden.

(c) Hier wird der Wert der Variablen verändert. Genauer gesagt, wird er um 3 erhöht.

Aufgabe 2. Der aktuelle Wert steht jeweils im Kommentar:

```
int meineZahl;
meineZahl = 5;
meineZahl = meineZahl * 2;    // meineZahl=10
meineZahl = meineZahl - 1;    // meineZahl=9
```

Aufgabe 3. Der jeweils veränderte Wert steht im Kommentar:

```
int meineZahl;
meineZahl = 5;
int meineZweiteZahl;
meineZweiteZahl = 10;
meineZahl = meineZahl * meineZweiteZahl; // meineZahl=50
meineZahl = meineZahl - 10;    // meineZahl=40
meineZweiteZahl = meineZweiteZahl + meineZahl;    // meineZweiteZahl=50
```

Aufgabe 4. Hier sind auch alternative Umsetzungen denkbar.

(a)

```
int klassenkasse;

klassenkasse = 0; // zu Beginn leer
klassenkasse = klassenkasse + 27*5; // 27 Schueler zahlen ein
klassenkasse = klassenkasse + 3*6; // 3 Nachzuegler
klassenkasse = klassenkasse - 30; // Bastelmaterial wird gekauft
klassenkasse = klassenkasse - 60; // Kosten fuer das Schwimmbad
```

(b)

```
// Jeder erhaelt so viel:
System.out.println(klassenkasse / 30);

// So viel bleibt uebrig:
System.out.println(klassenkasse % 30);
```

(c)

```
System.out.println(klassenkasse / 30.);
```

2.2 Variablen vergleichen

Aufgabe 5. Man erhält die Ausgaben `true`, `false`, `false`, `true`. Es wird nacheinander geprüft, ob meineZahl kleiner ist als deineZahl, ob meineZahl größer ist als deineZahl, ob meineZahl gleich 8 ist und ob meineZahl gleich 5 ist. Das Ergebnis der Prüfung kann entweder wahr oder falsch sein.

Aufgabe 6.

Vergleichsoperator	Prüfung
==	Sind beide Werte gleich?
!=	Sind beide Werte ungleich?
<	Ist der erste Wert kleiner?
>	Ist der erste Wert größer?
<=	Ist der erste Wert kleiner als oder gleich dem zweitem?
>=	Ist der erste Wert größer als oder gleich dem zweitem?

Aufgabe 7.

(a) Im Kommentar der jeweils aktuelle Wert der veränderten Variable:

```
int spielerEins;
int spielerZwei;

spielerEins = 0;
spielerZwei = 0;

spielerEins = spielerEins + 10; // 10
spielerZwei = spielerZwei + 5; // 5

spielerEins = spielerEins - 3; // 7
spielerZwei = spielerZwei*2; // 10

spielerEins = spielerEins + 11; // 18
spielerZwei = spielerZwei - 1; // 9

spielerEins = spielerEins + spielerZwei; // 27

spielerEins = spielerEins - 9; // 18
spielerZwei = spielerZwei + 9; // 18
```

(b) Hier kann man diesen Vergleich benutzen

```
System.out.println(spielerEins == spielerZwei);
```

und stellt prompt fest, dass beide gleich viele Punkte haben.

(c) Dies lässt sich so prüfen:

```
System.out.println(spielerEins > 20);
```

Aufgabe 8. (a)

```
double ersteZeit;
double zweiteZeit;
double dritteZeit;

ersteZeit = 10.1;
zweiteZeit = 10.5;
dritteZeit = 10.2;
```

(b)

```
double mittelwert;
mittelwert = (ersteZeit+zweiteZeit+dritteZeit)/3;

System.out.println(mittelwert);
```

(c)

```
double mittelwertZweiter;
mittelwertZweiter = (9.9+10.3+10.4)/3;

System.out.println(mittelwert == mittelwertZweiter);
System.out.println(mittelwert < mittelwertZweiter);
```

Man stellt fest, dass der zweite Mittelwert kleiner ist.

Zusatzaufgabe 1. Es ist nicht möglich, den Wert $0,5$ in einer Variablen vom Typ **int** zu speichern. Entsprechend erhält man eine Meldung, dass eine solche Typkonvertierung nicht möglich ist.

Im zweiten Test erhält man die Ausgabe `7.0`, woran man erkennt, dass die Zahl von Java als eine Dezimalzahl behandelt wird.

Zusatzaufgabe 2. Die Versuche der Schüler sollten zeigen, dass es bei den Vergleichen keine Probleme gibt. Allenfalls Rundungsfehler könnten hier für Überraschungen sorgen. Inwiefern man dies problematisiert sollte man von Kurs abhängig machen.

Zusatzaufgabe 3. Hier ist die Kreativität der Schüler gefragt.

SCHOOL-SCOUT.DE

Unterrichtsmaterialien in digitaler und in gedruckter Form

Auszug aus:

Einführung in Java - Lehrerband

Das komplette Material finden Sie hier:

School-Scout.de