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 3

Abgabe: 13.11.2014, 12:00 — Übung: 17./18.11.2014

Hausaufgabe 3: (4 Theoriepunkte)

Schreiben Sie ein Programm zur Überprüfung von ISBN-Nummern, und zwar den klassischen 10-stelligen Nummern (ISBN-10) wie z.B. `0-321-34980-6` (das ist die Nummer von "The Java Programming Language, Fourth Edition"). Die Striche sind nicht relevant und können weggelassen werden. Die Eingabe für Ihr Programm wäre also einfach "`0321349806`". Wenn Sie wollen, können Sie Eingaben mit Strichen erlauben, und die Striche einfach ignorieren. Das ist aber keine Bedingung, und es gibt keinen Extrapunkt dafür.

Die Prüfziffer am Ende kann außer den üblichen Ziffern 0 bis 9 auch ein "`X`" sein, das für die Ziffer 10 steht (eine römische Ziffer). Wenn man 11 mögliche Prüfziffern zur Verfügung hat, kann man mehr Fehler erkennen. Bei ISBNs kann eine falsche Ziffer oder Vertauschungen zweier nebeneinander stehender Ziffern erkannt werden. Z.B. ist auch "`0-201-15790-X`" eine mögliche ISBN (eingegeben als "`020115790X`").

Eine ISBN-10 besteht aus 9 Ziffern und Prüfziffer. Man berechnet die Prüfziffer, indem man die Zahlwerte jeder Ziffer mit ihrer Position multipliziert, alles aufsummiert, und dann den Rest bei Division durch 11 nimmt. Beispiel: `020115790X`. Man berechnet also

$$\underline{0}*1+\underline{2}*2+\underline{0}*3+\underline{1}*4+\underline{1}*5+\underline{5}*6+\underline{7}*7+\underline{9}*8+\underline{0}*9 = 0+4+0+4+5+30+49+72+0 = 164$$

Dies teilt man durch 11 mit Rest, gibt 14 Rest 10. Weil der Rest 10 ist, ist die Prüfziffer `X`.

Ihr Programm soll folgende Ausgaben haben:

- "`Ok`", falls die ISBN korrekt ist,
- "`Laenge falsch`", falls die Eingabe nicht 10 Zeichen sind (ggf. nach der optionalen Eliminierung des "-"),
- "`ungueltiges Zeichen:` ...", falls eines von den ersten neun Zeichen keine Dezimalziffer ist (geben Sie nur das erste falsche Zeichen aus),
- "`Pruefziffer inkorrekt, richtig waere` ...", falls die Prüfziffer (zehntes Zeichen) falsch ist. (Geben Sie nur die korrekte Prüfziffer aus, nicht die die ganze ISBN. Die Information ist nützlich, wenn man z.B. zu einer ISBN-13 eine ISBN-10 berechnen will. Die Prüfziffern in beiden Systemen sind unterschiedlich.)

Lesen Sie die Eingabe als Zeichenkette ein, z.B. mit der Methode `next()` der `Scanner`-Klasse:

```
// Hausaufgabe 3: ISBN-Codes
// Name: (bitte hier Ihren Namen eintragen)

import java.util.Scanner;

class ISBN {
        public static void main(String[] args) {
                Scanner scan = new Scanner(System.in);
                System.out.print("Bitte ISBN eingeben: ");
                String isbn = scan.next();

                ...
        }
}
```

Die Länge eines Strings `s` können Sie mit `s.length()` bestimmen, das `i`-te Zeichen erhalten Sie mit `s.charAt(i)`. Dies wurde in der Vorlesung besprochen, bei Bedarf schauen Sie in die Dokumentation:

http://docs.oracle.com/javase/7/docs/api/java/lang/String.html

Beachten Sie, dass Sie die Ziffern als Zeichen vom Typ `char` klar unterscheiden müssen von ihren Zahlwert. Z.B. hat die Ziffer `'0'` in den Code 48 (in ASCII bzw. Unicode). Um zu prüfen, ob ein Zeichen `c` eine Ziffer ist, können Sie `Character.isDigit(c)` verwenden. Es handelt sich um eine statische Methode der Bibliotheks-Klasse `Character`, dokumentiert unter

http://docs.oracle.com/javase/7/docs/api/java/lang/Character.html

Alternativ könnten Sie auch testen, ob `c >= '0' & c <= '9'`. Die Umwandlung einer Ziffer `c` in den entsprechenden Zahlwert geht mit `Character.digit(c, 10)`. Alternativ können Sie auch einfach `c - '0'` rechnen.

Schreiben Sie das Programm in die Datei "`ISBN.java`". Geben Sie diese Datei über die Übungsplattform bis zum Donnerstag, 13.11.2014, 12:00 ab. Beachten Sie, dass nur compilierbare Programme akzeptiert werden. Wir verwenden eine automatische Vorkorrektur, deswegen halten Sie sich bitte genau an die Vorgaben. Bitte achten Sie auch auf sinnvolle Einrückungen und allgemein ein verständliches Programm. Für schlechten Programmierstil können auch Punkte abgezogen werden.

Übungsaufgabe 3A: (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.

Im vierten Kapitel der Vorlesung

`http://users.informatik.uni-halle.de/~brass/oop14/long_print/j4_lexsy.pdf`

steht auf Folie 51 der Syntaxgraph zu "DecimalFloatingPointLiteral". Prüfen Sie anhand dieses Syntaxgraphen, ob folgende Konstanten von Typ `double` legal sind:

a) `00.e-00`

b) `-1.E+20`

c) `15D`

d) `239`

e) `.3456`

f) `0e1.0`

Übungsaufgabe 3B: (ohne Abgabe)

Was gibt das folgende Programm aus? Versuchen Sie das Ergebnis zu bestimmen, ohne das Programm auszuführen.

```
1  class Aufg3B {
2          public static void main(String[] args) {
3                  StringBuilder b = new StringBuilder("ABCDE");
4                  b.setCharAt(1, 'X');
5                  StringBuilder c = b;
6                  c.setCharAt(3, 'X');
7                  int i = b.length() - 1;
8                  while(i >= 0) {
9                          if(b.charAt(i) == 'X')
10                                 System.out.println(i);
11                         i = i - 1;
12                 }
13                 System.out.println(c);
14                 System.out.println(i);
15         }
16 }
```

Was würde passieren, wenn man nach dem `if(...)` in Zeile 9 ein Semikolon schreibt? (Hinweis: Eine Anweisung in Java darf leer sein, "`;`" ist also eine gültige Anweisung.)

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