

Die Klasse Scanner

Mit einem Objekt der Klasse `java.util.Scanner` kann man Werte verschiedener Typen (boolean, byte, short, int, long, float, double, BigInteger, BigDecimal und weitere) aus einer Datei, einem Strom oder einem String-Objekt einlesen (z.B. von der Standard-Eingabe `System.in`). Es folgen hier zwei typische Anwendungen mit Scanner-Objekten.

```

1 // Datei Scanner01.java
2 /* -----
3 Mit Hilfe eines Scanner-Objekts kann man z.B. int-Werte, float-Werte,
4 boolean-Werte oder Strings von der Standardeingabe einlesen.
5 Mit den Methoden hasNextInt, hasNextFloat, hasNextBoolean etc. kann man
6 pruefen, ob der naechste Teil der Eingabe als int-Wert, float-Wert oder
7 boolean-Wert etc. interpretiert werden kann.
8 Mit den Methoden nextInt, nextFloat, nextBoolean etc. kann man dann einen
9 Wert des betreffenden Typs einlesen.
10
11 Die Klasse Scanner wurde (erst, endlich) mit Java 5 eingefuehrt.
12 ----- */
13 import java.util.Scanner;
14
15 class Scanner01 {
16     // -----
17     static public void main(String[] _) {
18         printf("Scanner01: Jetzt geht es los!\n");
19         printf("-----\n");
20
21         Scanner scin = new Scanner(System.in);
22
23         while (true) {
24             printf("Eine Eingabe (q zum Beenden): ");
25             if (scin.hasNextInt()) {
26                 int i = scin.nextInt();
27                 printf("Das war eine Ganzzahl           : %d\n", i);
28             } else if (scin.hasNextFloat()) {
29                 float f = scin.nextFloat();
30                 printf("Das war eine Bruchzahl           : %.2f\n", f);
31             } else if (scin.hasNextBoolean()) {
32                 boolean b = scin.nextBoolean();
33                 printf("Das war ein Wahrheitswert       : %b\n", b);
34             } else if (scin.hasNext()) {
35                 String s = scin.next();
36                 printf("Das war ein String                 : %s\n", s);
37                 if (s.equals("q")) break;
38             }
39         }
40
41         printf("-----\n");
42         printf("Scanner01: Das war's erstmal!\n");
43     } // main
44     // -----
45     // Eine Methode mit einem kurzen Namen:
46     static void printf(String f, Object... v) {System.out.printf(f, v);}
47     // -----
48 } // class Scanner01
49 /* -----
50 Ein Dialog mit diesem Programm (Eingaben des Benutzer fett hervorgehoben):
51
52 Scanner01: Jetzt geht es los!
53 -----
54 Eine Eingabe (q zum Beenden): 888888888
55 Das war eine Ganzzahl           : 888888888
56 Eine Eingabe (q zum Beenden): 8888888888
57 Das war eine Bruchzahl           : 88888889344,00

```

```

58 Eine Eingabe (q zum Beenden): 1,235
59 Das war eine Bruchzahl      : 1,24
60 Eine Eingabe (q zum Beenden): 2,235
61 Das war eine Bruchzahl      : 2,23
62 Eine Eingabe (q zum Beenden): 1,0e38
63 Das war eine Bruchzahl      : 99999996802856920000000000000000000000,00
64 Eine Eingabe (q zum Beenden): 1,0e39
65 Das war eine Bruchzahl      : Infinity
66 Eine Eingabe (q zum Beenden): NaN
67 Das war eine Bruchzahl      : NaN
68 Eine Eingabe (q zum Beenden): nan
69 Das war ein String          : nan
70 Eine Eingabe (q zum Beenden): true
71 Das war ein Wahrheitswert   : true
72 Eine Eingabe (q zum Beenden): False
73 Das war ein Wahrheitswert   : false
74 Eine Eingabe (q zum Beenden): Hallo!
75 Das war ein String          : Hallo!
76 Eine Eingabe (q zum Beenden): "Hallo Sonja"
77 Das war ein String          : "Hallo
78 Eine Eingabe (q zum Beenden): Das war ein String          : Sonja"
79 Eine Eingabe (q zum Beenden): q
80 Das war ein String          : q
81 -----
82 Scanner01: Das war's erstmal!
83 -----
84 Erlaeuterungen:
85 1. Die Zahl 8888888888 (zehn achten) ist groesser als der groesste
86   int-Wert (2147483648) und wird von diesem Programm deshalb als
87   float-Wert eingelesen. Der "naechste float Wert zu 8888888888"
88   ist allerdings um 456 zu gross.
89 2. Die Bruchzahl 1,235 wird bei der Ausgabe zu 1,24 gerundet, aber
90   2,235 wird nicht aufgerundet. Regel: Es kommt auf den genauen
91   float-Wert an, nicht auf seine ungefaehre dezimale Darstellung!
92 3. Die Bruchzahl 1,0e38 kann natuerlich nur naeherungsweise als
93   float-Wert dargestellt werden. Der von Java verwendete Naehierungs-
94   wert ist allerdings um etwa 3 Nonillionen (eine 3 gefolgt von
95   30 Nullen) zu klein. Als absolute Zahl betrachtet ist dieser
96   Fehler ziemlich gross. Als relativer Fehler ist er trotzdem ziemlich
97   klein: weniger als ein Promille von einem Promille. In vielen
98   Zusammenhaengen kann man relative Fehler dieser Groessenordnung
99   vernachlaessigen.
100 4. Die Bruchzahl 1,0e39 ist groesser als der groesste float-Wert
101   (etwa 3,402e38) und wird deshalb durch den speziellen float-Wert
102   Infinity (unendlich) dargestellt.
103 5. Eine NaN ("not a number") ist ein float-Wert, der keine Zahl darstellt.
104   Zum Typ float gehoeren etwa 16,8 Millionen NaNs (das sind ungefaehr
105   3,9 Promille aller float-Werte). Die uebrigen float-Werte (996,1 Pro-
106   mille aller float-Werte) stellen Zahlen dar.
107 6. Wenn man eine NaN eingeben will, muss man auf GROSS/klein-Schreibung
108   achten: "NaN" wird als float-Eingabe akzeptiert, "nan" und "NAN" etc.
109   nur als String-Eingaben.
110 7. Wenn man die Wahrheitswerte true und false eingeben will, braucht
111   man nicht auf GROSS/klein-Schreibung zu achten.
112 8. Beim Einlesen eines Strings mit der Methode getNext eines Scanner-
113   Objekts beendet das erste transparente Zeichen (engl. white space
114   character, hier Blanks und Tab-Zeichen) den String. Das bedeutet:
115   Mit getNext kann man nicht ohne weiteres einen String einlesen, der
116   transparente Zeichen enthaelt. Auch mit doppelten Anfuhrungszeichen
117   kann man daran nichts aendern.
118 ----- */

```

```

1 // Datei Scanner03.java
2 /* -----
3 Mit einem Scanner-Objekt kann man Daten nicht nur von der Standardeingabe
4 (System.in), sondern aus einem beliebigen Readable-Objekt oder aus einem
5 String-Objekt lesen.
6
7 Ausserdem kann man Ganzzahlen nicht nur 10-er-Zahlen (Dezimalzahlen),
8 sondern auch 2-er-Zahlen, 3-er-, 4-er-, ..., 36-er-Zahlen einlesen.
9
10 Als Zeichen zur Darstellung von Zahlen wollte man nur die 10 Dezimalziffern
11 und die 26 Buchstaben verwenden (denn mit deren Reihenfolge sinde viele
12 Menschen vertraut). Deshalb ist die groesste zulaessige Basis gleich 36.
13 ----- */
14 import java.util.Scanner;
15
16 class Scanner03 {
17     // -----
18     static public void main(String[] sonja) {
19         printf("Scanner03: Jetzt geht es los!\n");
20         printf("-----\n");
21         printf("Character.MIN_RADIX: %2d\n", Character.MIN_RADIX);
22         printf("Character.MAX_RADIX: %2d\n", Character.MAX_RADIX);
23         printf("-----\n");
24
25         // Jeder Eingabestring enthaelt eine Basis (als 10-er-Zahl notiert)
26         // und eine Zahl im System mit dieser Basis.
27         // Als Zusaetze snd erlaubt:
28         // ein Vorzeichen Minus '-' vor der ersten Ziffer und
29         // Trennzeichen zwischen 3-er-Gruppen von Ziffern (in DE: '.')
30         String[] eingabe = {
31             " 2      101",
32             " 2 -11.111.111",
33             " 5       40",
34             " 5      -1.100",
35             "10      789",
36             "10     -12.345",
37             "16      100",
38             "16     -1.000",
39             "20       j",
40             "20      -J0",
41             "36       z",
42             "36     -20",
43         };
44
45         // Ein paar Ueberschriften ausgeben:
46         printf("%-15s  %-10s\n", "Eingabe-", "Als"      );
47         printf("%-15s  %-10s\n", "String:", "10-er-Zahl:");
48         printf("\n");
49
50         // Jeden Eingabestring mit einem Scanner-Objekt als int-Wert
51         // "einlesen" und diesen Wert als 10-er-Zahl wieder ausgeben:
52         for (String ein : eingabe) {
53             Scanner sc = new Scanner(ein);
54             int     basis = sc.nextInt();
55             int     zahl = sc.nextInt(basis);
56             printf("\n%15s\  %10d\n", ein, zahl);
57         }
58         printf("-----\n");
59         printf("Scanner03: Das war's erstmal!\n");
60     } // main
61     // -----
62     // Eine Methode mit einem kurzen Namen:
63     static void printf(String f, Object... v) {System.out.printf(f, v);}
64     // -----
65 } // class Scanner03
66 /* -----

```

Ausgabe:

```
67
68 Scanner03: Jetzt geht es los!
69 -----
70 Character.MIN_RADIX: 2
71 Character.MAX_RADIX: 36
72 -----
73 Eingabe-           Als
74 String:           10-er-Zahl:
75
76 "  2           101"           5
77 "  2 -11.111.111"         -255
78 "  5           40"           20
79 "  5      -1.100"         -150
80 " 10           789"          789
81 " 10      -12.345"        -12.345
82 " 16           100"          256
83 " 16      -1.000"         -4.096
84 " 20           j"           19
85 " 20          -j0"         -380
86 " 36           z"           35
87 " 36          -20"         -72
88 -----
89 Scanner03: Das war's erstmal!
90 ----- */
```

Die Dateien Scanner01.java bis Scanner03.java findet man im Archiv BspJaSp.zip auf der Netzseite <http://public.beuth-hochschule.de/~grude/JavaIstEineSprache/welcome.html>.