

Zufallszahlen in Java

== Beispiel für eine Zufallsklasse ==

Folgendes Beispiel zeigt, wie man in Java an zufällig generierte Zahlen kommen kann:

Quellcode

```
1. //this is the package that belongs to this class
2. //you are free to change it to anything else
3. package TWSargTeX;
4. //this command is very important
5. //it imports the required Random-class from the java-utils
6. import java.util.Random;
7. /**
8.  * This class creates random numbers
9.  *
10. * @author Martin Bories
11. * @copyright 2009 Martin Bories, all rights reserved
12. * @license: You are allowed to use this free and to change the copyright!
13. */
14. public class RandomNumbers {
15.     /**
16.      * @access public
17.      * @method getRandomNumber()
18.      * @return numberIntegercontains a random created number as an integer
19.      */
20.     public static int getRandomNumber() {
21.         Random randomGenerator = new Random();
22.         return randomGenerator.nextInt();
23.     }
24.     /**
25.      * this function is a synonym to the "getRandomNumber()"-function
26.      *
27.      * @access public
28.      * @method getRandomInt()
29.      * @return numberIntegercontains a random created number as an integer
30.      */
31.     public static int getRandomInt() {
32.         Random randomGenerator = new Random();
33.         return randomGenerator.nextInt();
34.     }
35.     /**
36.      * this function returns a random created double
37.      *
38.      * @access public
39.      * @method getRandomDouble()
40.      * @return numberDoublecontains a random created number in the double format
41.      */
42.     public static double getRandomDouble() {
43.         Random randomGenerator = new Random();
44.         return randomGenerator.nextDouble();
45.     }
46.     /**
47.      * this function returns a random created float
48.      *
49.      * @access public
50.      * @method getRandomFloat()
```

```
57. * @return numberFloatcontains a random created number in the float format
58. */
59. public static float getRandomFloat() {
60. Random randomGenerator = new Random();
61. return randomGenerator.nextFloat();
62. }
63. }
```

Alles anzeigen

Hier ist die Datei zum kostenfreien Download: [RandomNumbers.java](#)

== Zufallswerte selbst erzeugen ==

Generell gibt es die Klasse "java.util.Random", welche Zufallswerte generiert. Diese muss instanziiert werden

Quellcode

```
1. Random randomGenerator = new Random();
```

bevor sie verwendet werden kann. Danach weist die Instanz verschiedene Funktionen auf, wie zum Beispiel

Quellcode

```
1. //gibt einen zufälligen Integer-Wert zurück
2. Random.nextInt();
3. //gibt einen zufälligen Float-Wert zurück
4. Random.nextFloat();
5. //gibt einen zufälligen Double-Wert zurück
6. Random.nextDouble();
7. //gibt einen zufälligen Boolean-Wert zurück
8. Random.nextBoolean();
9. //gibt einen zufälligen Long-Wert zurück
10. Random.nextLong();
```

== Zufallswerte beschränken ==

Bei der Funktion Random.nextInt() lässt sich zudem ein Bereich zwischen 0 und n setzen, aus dem Zufallszahlen generiert werden:

Quellcode

```
1. //gibt einen zufälligen Int-Wert zwischen 0 und 10 zurück
2. Random.nextInt(10);
3. //gibt einen Error aus, da n > 0 sein muss
4. Random.nextInt(0);
```

Wichtig dabei ist (wie das letzte Beispiel zeigt), dass n größer als 0 sein muss- sonst gibt es einen Error. Es empfiehlt sich also, bei Begrenzung des Bereiches vorher zu Kontrollieren, ob n > 0 ist:

Quellcode

```
1. //dies ist nur in einer schleife oder bei einer eingabe des benutzers notwendig, da n sonst vom programmierer
2. //auf einen Wert > 0 gesetzt werden sollte
3. if (n > 0) {
4. Random.nextInt(0);
5. }
```

== Fazit ==

Ich hoffe, dass dieses kleine Tutorial ein paar Leuten helfen konnte :).

== Quellen ==

- [SargTeX.de- Zufallszahlen in Java](#)