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
 package TWSargTeX;
 5. //this command is very important
 6. //it imports the required Random-class from the java-utils
 7. import java.util.Random;
 9. /**
10. * This class creates random numbers
11. *
12. * @author Martin Bories
13. * @copyright 2009 Martin Bories, all rights reserved
14. * @license: You are allowed to use this free and to change the copyright!
15. */
public class RandomNumbers {
18. /**
19. * @access public
20. * @method getRandomNumber()
21. * @return numberIntegercontains a random created number as an integer
22. */
23. public static int getRandomNumber() {
24. Random randomGenerator = new Random();
25. return randomGenerator.nextInt();
26. }
28. /**
29. * this function is a synonym to the "getRandomNumber()"-function
30. *
31. * @access public
32. * @method getRandomInt()
33. * @return numberIntegercontains a random created number as an integer
34. */
35. public static int getRandomInt() {
Random randomGenerator = new Random();
37. return randomGenerator.nextInt();
38. }
39. /**
41. * this function returns a random created double
42. *
43. * @access public
44. * @method getRandomDouble()
45. * @return numberDoublecontains a random created number in the double format
46. */
47. public static double getRandomDouble() {
48. Random randomGenerator = new Random();
49. return randomGenerator.nextDouble();
50. }
52. /**
53. * this function returns a random created float
54. *
55. * @access public
56. * @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

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
- Random.nextDouble();
- 7. //gibt einen zufälligen Boolean-Wert zurück
- 8. Random.nextBoolean();
- 9. //gibt einen zufälligen Long-Wert zurück
- 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
- Random.nextInt(10);
- 3. //gibt einen Error aus, da n > 0 sein muss
- 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