



Kostenloses eBook

LERNEN

random

Free unaffiliated eBook created from
Stack Overflow contributors.

#random

Inhaltsverzeichnis

Über	1
Kapitel 1: Erste Schritte mit Zufall	2
Bemerkungen.....	2
Examples.....	2
Installation oder Setup.....	2
Fisher-Yates mischen.....	2
Credits	4



You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [random](#)

It is an unofficial and free random ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official random.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Kapitel 1: Erste Schritte mit Zufall

Bemerkungen

In diesem Abschnitt erhalten Sie einen Überblick darüber, was zufällig ist und warum ein Entwickler es verwenden möchte.

Es sollte auch alle großen Themen in zufälliger Reihenfolge erwähnen und auf die verwandten Themen verweisen. Da die Dokumentation für random neu ist, müssen Sie möglicherweise erste Versionen dieser verwandten Themen erstellen.

Examples

Installation oder Setup

Detaillierte Anweisungen zum Einrichten oder Installieren der Zufallsdaten.

Fisher-Yates mischen

Auch bekannt als Knuth-Shuffle und Durstenfeld-Fisher-Yates-Shuffle. Dieser Shuffle nimmt ein Array von n Elementen und mischt es. Der Algorithmus ist wirklich zufällig, da nach dem Mischen jede Permutation des Arrays gleich wahrscheinlich ist.

In Java:

```
public static void shuffle(E[] deck) {  
  
    //From the end, swap each card with a random card from the unswapped portion.  
    for(int i = deck.length - 1; i > 0; i--)  
    {  
        //Pick an element from [0,i], inclusive.  
        int chosenCard = (int) (Math.random() * (i + 1));  
  
        E temp = deck[i];  
        deck[i] = deck[chosenCard];  
        deck[chosenCard] = temp;  
    }  
}
```

Bitte beachten Sie: Es ist notwendig, dass das Ersetzungselement aus $[0, i]$ einschließlich und nicht aus $[0, i)$ stammt. Andernfalls sind Permutationen des Arrays, bei denen Elemente bestehen bleiben, unmöglich, was nicht wirklich zufällig ist.

Unter der Annahme, dass die Erzeugung von Zufallszahlen $O(1)$ erfordert, arbeitet der Algorithmus an Ort und Stelle und benötigt $O(n)$ Zeit und Raum. Ein auf diese Weise umgeordnetes Array kann verwendet werden, um sich nicht wiederholende Elemente in $O(1)$ amortisierter Zeit pro Element abzurufen.

```
E[] deck;
int drawIndex;

//Elements are taken from an index that advances.
public E drawUniqueCard()
{
    //Once all cards have been drawn, reshuffle the deck and draw from the top.
    if(drawIndex == deck.length)
    {
        shuffle(deck);
        drawIndex = 0;
    }
    //Pull the next card off the deck.
    return deck[drawIndex++];
}
```

Erste Schritte mit Zufall online lesen: <https://riptutorial.com/de/random/topic/9484/erste-schritte-mit-zufall>

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit Zufall	Community , Mauve Ranger