

# Programmieren C++: Templates, STL vector, File-I/O, Exceptions: File kopieren mit zufälligem Umordnen der Zeilen

## Klaus Kusche

Wir wollen ein Programm schreiben, das eine angegebene Datei zeilenweise komplett in den Speicher liest, dann die Zeilen zufällig vertauscht, und schließlich die umgeordneten Zeilen in eine andere angegebene Datei speichert.

Zum internen Speichern der eingelesenen Zeilen verwenden wir ein Objekt der vordefinierten STL-Container-Klasse vector mit Elementtyp **string**. Vektoren verhalten sich in vieler Hinsicht wie Arrays (man kann z.B. mit [ ] auf die einzelnen Elemente zugreifen), aber sie wachsen automatisch nach Bedarf, d.h. man muss beim Anlegen keine Größe angeben.

Das ist in diesem Beispiel besonders praktisch, da wir ja vorab nicht wissen, wie viele Zeilen die einzulesende Datei enthält.

### a) Funktion scramble

Die Funktion **scramble** bekommt als Parameter einen **vector** mit beliebigem Element-Typ (die Funktion ist daher als Template zu programmieren). Wie musst du diesen Vektor-Parameter übergeben, damit die Funktion die Elemente des Original-Vektors umordnen kann? Die Funktion hat keinen Returnwert.

Sie soll alle Elemente dieses Vektors zufällig umordnen, und zwar nach dem Fisher-Yates-Verfahren:

- Schreibe eine Schleife, die mit **i** vom Index des letzten Elementes des Vektors rückwärts bis **1** zählt (die Methode **size** liefert dir die Größe eines Vektors).
- Ermittle in der Schleife eine zufällige Position pos zwischen **0** und **i** (jeweils einschließlich!).
- Vertausche das Vektor-Element an Position **pos** mit dem Element an Position **i** (du darfst das vordefinierte **swap** aus **utility** zum Vertauschen verwenden).

### b) Funktion copy\_scrambled

Das Einlesen und Ausgeben soll in einer Funktion **copy\_scrambled** geschehen. Diese Funktion hat zwei string-Parameter (die Dateinamen der zu lesenden und der zu schreibenden Datei) und keinen Returnwert.

- Öffne zuerst die beiden Dateien.  
Wirf einen **string** mit einem Fehlertext als Exception, wenn das Öffnen fehlschlägt.
- Deklariere einen **vector** mit **string**-Elementen als lokale Variable.
- Lies in einer Schleife immer wieder eine Zeile in eine **string**-Hilfsvariable (mit der Funktion **getline**, nicht mit der Methode **getline**) und hänge die gelesene Zeile mit der Methode **push\_back** hinten an den Vektor an, bis das Lesen einen Fehler liefert.

- Ruf unsere Funktion **scramble** für den Vektor auf.
- Schreib mit einer weiteren Schleife alle Elemente des Vektors in die Ausgabedatei (schaffst du das auch mit einer **for** ( ... : ... ) -Schleife?).

### c) Hauptprogramm

Unser Hauptprogramm soll mit zwei Argumenten auf der Befehlszeile aufgerufen werden: Dem Dateinamen der Eingabedatei und dem Dateinamen der Ausgabedatei.

- Prüfe zuerst, ob wirklich zwei Namen auf der Befehlszeile angegeben wurden. Wenn nicht: Fehlermeldung, Programmende.
- Sorge dafür, dass das Programm bei jedem Programmlauf eine andere Vertauschung der Zeilen liefert.
- Ruf **copy\_scrambled** auf. Falls es eine Fehlermeldung wirft: Fange sie, gib sie aus, und beende das Programm.

### Zusatzaufgabe

Es gibt in **<algorithm>** bereits eine fertige Funktion **random\_shuffle** zum Permutieren eines Containers mittels Fisher-Yates.

Versuche, den Aufruf deiner Funktion durch diese Funktion zu ersetzen!