

Programmieren C++: Files, Stringstreams

Klaus Kusche

Stringstream-Eingabe: Hex- und Oktal-Zahlen einlesen

Gesucht ist ein Programm, das formatierte **int**-Zahlen zeilenweise (eine Zahl pro Zeile) von einem File (Name auf der Befehlszeile angegeben) nach **cout** kopiert. Die Zahlen in der Eingabe können dezimal, hexadezimal oder oktal sein, ausgegeben werden sollen sie in allen Fällen dezimal.

Das Format jeder einzelnen Eingabezahl ist am letzten Zeichen der jeweiligen Zeile zu erkennen: Endet die Zeile auf "h" oder "H", enthält sie eine Hexzahl, endet sie auf "o" oder "O", steht davor eine Oktalzahl, und ist beides nicht der Fall, wird davon ausgegangen, dass die Zahl in dieser Zeile eine Dezimalzahl ist.

Verwende folgende Strategie zur Lösung der Aufgabe:

- Der auf der Befehlszeile angegebene File wird zum Lesen geöffnet (prüfe, ob ein Filename angegeben ist, und prüfe das Öffnen auf Fehler!).
- In einer Schleife geschieht pro Zeile Folgendes:
 - Die Eingabe wird mittels **getline** zeilenweise vom Eingabefile in einen String (einen C-String, d.h. ein **char**-Array, keinen C++-String) gelesen. **getline** ist eine Methode der Inputfile-Klasse, sie hat den zu befüllenden String und seine maximale Länge als Parameter.
 - Mit diesem String wird ein Stringstream angelegt. Du brauchst für jede Zeile einen neuen Stringstream, also deklarier ihn erst in der Schleife und nicht davor! Gib gleich in der Deklaration als String des Stringstreams die gelesene Zeile an!
 - Dann wird das letzte Zeichen des Strings geprüft (berechne dessen Index mit **strlen** aus **<cstring>**).

Je nachdem, ob es 'h'/'H', 'o'/'O' oder etwas anderes ist (**tolower** aus **<cctype>** ist besser als zwei **if**-Bedingungen für Groß- & Kleinbuchstabe!), wird aus dem Stringstream eine Hexzahl, eine Oktalzahl oder eine normale Zahl in einen **int** gelesen.

Das Einlesen einer **int**-Zahl **zahl** in Hex, Dezimal oder Oktal funktioniert so:

```
... >> hex >> zahl; ... >> dec >> zahl; ... >> oct >> zahl;
```

Zusatzaufgabe: Prüfe nach dem Lesen aus dem Stringstream mit **.fail()** auf Konvertierungsfehler, z.B. wenn die Zeile gar keine Zahl enthielt!

- Der gelesene **int**-Wert wird ganz normal mit << auf **cout** ausgegeben.

Die **while**-Schleife für die Zeilen soll enden, wenn beim **getline** End of File oder irgendein Fehler auftritt (das **getline** returniert den File und kann direkt als Schleifen-Bedingung verwendet werden), oder wenn die gelesene Zeile Länge 0 hat (dafür kannst du sinnvollerweise ein **break** verwenden).