

Inf Programmieren 1 Übung: Template-Klassen

Klaus Kusche

Buchstaben und Worte zählen

Wir wollen ein Programm schreiben, das zählt, wie oft jeder Buchstabe und wie oft jedes Wort in einer Datei vorkommt. Weiters wird gezählt, wie oft jede Wortlänge vorkommt.

Alle drei Zählungen sollen durch verschiedene Instanzen derselben Template-Klasse geschehen: Einmal zählt die Klasse Werte vom Typ **char** (nicht **char ***, sondern wirklich einzelne Buchstaben), einmal vom Typ **string** (C++ Strings, nicht C Strings), und einmal vom Typ **int**.

Das Template hat also den Typ der zu zählenden Werte als Parameter.

Das Hauptprogramm liegt der Angabe bei, es muss nur das Template ergänzt werden.

Hinweise zur Template-Klasse:

- Die maximale Anzahl **size** der verschiedenen Werte, die die Klasse zählen kann, soll ebenfalls Parameter des Templates sein: Bei den Buchstaben werden das nur 30 sein (das Hauptprogramm zählt nur die "echten" Buchstaben, keine Zahlen oder Sonderzeichen), bei den Längen sollten 50 reichen, bei den Worten sind mehr nötig (z.B. 1000).
- Die Klasse hat 3 Member: Ein Array vom zu zählenden Typ mit **size** Elementen für die Werte, ein **int**-Array mit **size** Elementen für die dazugehörigen Zähler (d.h. der **i**-te Zähler gehört zum **i**-ten Wert), und einen einzelnen **int**, der angibt, wie viele Elemente in den beiden Arrays schon belegt sind.
- Der Konstruktor hat keinen Parameter und setzt nur die Anzahl der bisher belegten Elemente auf 0 (die Arrays bleiben uninitialized).
- Außer dem Konstruktor hat die Klasse zwei Methoden:
 - Eine Methode **count**: Sie bekommt einen Wert übergeben und zählt ihn:
 - Wenn der Wert schon im Werte-Array vorkommt, wird der dazugehörige Zähler um eins erhöht.
 - Wenn nicht, wird der Wert mit Zählerstand 1 zu beiden Arrays dazugefügt (Achtung: Ist noch Platz? Wenn nicht: Programm mit Fehlermeldung abbrechen!)

Die Methode hat keinen Returnwert.

- Eine Methode **print** ohne Argumente und Returnwert: Sie geht die Arrays durch und gibt jeden Wert samt Zählerstand zeilenweise auf **cout** aus.

Da beide Methoden etwas umfangreicher sind, solltest du sie nicht inline sondern außerhalb von **class** definieren!

Zusatzaufgaben:

- Die Lösung mit zwei getrennten Member-Arrays für Werte und Zähler ist nicht besonders elegant. Kannst du eine zweite Template-Klasse für Paare von einem Wert **val** (von beliebigem Typ) und dem dazugehörigen **int**-Zähler **cnt** einführen und in der Zähler-Klasse dann ein einziges Array solcher Paare verwenden?

Die Paar-Klasse braucht keine Methoden und keinen Konstruktor:

Die beiden Member dürfen **public** sein und direkt angesprochen werden, es reicht daher eine normale **struct** statt **class**.

- Kannst du das Einfügen neuer Werte in der Methode **count** so umbauen, dass die Werte im Array stets sortiert sind (damit die Ausgabe der Zähler sortiert nach Werten ist)?

Erinnere dich an das übliche Vorgehen: Array von hinten nach vorne durchlaufen und bestehende Elemente um einen Platz nach hinten verschieben, bis an der richtigen Stelle ein Loch für das neue Element frei ist (die richtige Stelle kann man sich gleich beim Suchen des Wertes merken, dann braucht man beim Verschieben nicht mehr die Werte der Elemente vergleichen!).

- Kannst du statt der Methode **print** eine globale Template-Funktion **operator<<** definieren, die dasselbe leistet (und den Aufruf im **main** entsprechend ändern)? **operator<<** soll ein **friend** unserer Zähler-Template-Klasse sein.