

# Programmieren C++: Template-Funktionen

**Klaus Kusche**

## Notenliste

Der Programmier-Neuling D. Epperl schreibt ein Programm zur Verwaltung der Programmiernoten seiner Klasse.

Bisher hat er 3 globale Arrays: Ein **string**-Array (C++ **string**, nicht C **char \***) für die Namen der Schüler, ein **int**-Array für deren Matrikelnummern, und ein **double**-Array für die Noten (du darfst diese Arrays in der Deklaration fix initialisieren, du musst keine Benutzereingabe dafür programmieren). Auch die Anzahl der Schüler (die Größe der 3 Arrays) hat er in einer globalen Konstante **anzahl** gespeichert.

(Dass diese Datendarstellung und dieser Programmierstil alles andere als optimal ist und diese Informationen wohl in einer Klasse Schueler viel besser aufgehoben wären, wollen wir jetzt einmal nicht diskutieren.)

Weiters hat er schon eine globale Funktion **print(int i)** geschrieben, die eine Zeile mit den schön formatierten Daten des i-ten Schülers (Name, Matrikelnummer, Note) aus den globalen Arrays heraus ausgibt (**i** ist der Index des Schülers in den 3 Arrays).

Jetzt schreibt er gerade am Hauptprogramm: Es soll 6 Funktionalitäten bieten, die der Benutzer durch Eingabe eines Buchstabens auswählen kann: Ausgabe einer nach Namen, Matrikelnummer oder Noten sortierten Liste aller Schüler, und Suche nach einem einzelnen Namen, einer Matrikelnummer oder einer Note. Beim Implementieren der 6 Zweige seines **switch** fällt ihm auf, dass je drei Zweige fast identen Code enthalten.

Mach es besser: Schreibe ein **main** und **print** wie oben beschrieben, aber verwende für die drei Such-Fälle eine einzige Such-Funktion (als Template). Auch die drei verschiedenen sortierten Listen sollen von einer einzigen Template-Funktion ausgegeben werden.

### Hinweise:

- Beide Funktionen haben jeweils ein einziges Argument: Das Array, in dem gesucht bzw. nach dem sortiert ausgegeben werden soll. Returnwert haben sie keinen.
- Die **switch**-Fälle sollen nur aus dem Funktionsaufruf und dem **break** bestehen.
- Die Such-Funktion liest einen Namen, eine Matrikelnummer oder eine Note vom Terminal ein und gibt alle passenden Einträge mittels **print** aus. Wird nichts gefunden, soll eine entsprechende Meldung kommen.

Achtung: Eine Note kann mehrmals vorkommen, suche bis zum Ende der Arrays!

- Die Mühe, die Arrays in der Listen-Funktion für die Ausgabe zu sortieren, sparen wir uns. Verwende stattdessen folgende Idee:

Um  $n$  Einträge sortiert auszugeben, geben wir  $n$  Mal den kleinsten bisher noch nicht ausgegebenen Eintrag aus (d.h. wir durchlaufen  $n$  Mal alle Einträge, um den kleinsten verbliebenen zu suchen). Wir verwenden dazu ein Hilfsarray mit  $n$  **bool**-Elementen, in dem wir uns merken, welche Einträge schon ausgegeben wurden.

Achtung: Um die Funktion **print** aufrufen zu können, musst du dir nicht den Wert des kleinsten Eintrags merken, sondern dessen Index!