

# Programmieren C++: Operatoren, Objekte mit dyn. Daten: Klasse für Messwerte

## Klaus Kusche

Gesucht ist eine Klasse **Messw**, deren Objekte jeweils ein Array von Messwerten speichern, wobei die einzelnen Werte Kommazahlen sind.

- Das Array soll dynamisch angelegt (und wieder freigegeben!) werden, das Objekt enthält einen Pointer darauf als Member-Variable.
- Das Array soll immer genau so groß angelegt werden, wie es gerade benötigt wird. Die Größe des Arrays (d.h. die Anzahl der Messwerte, die das **Messw**-Objekt gerade enthält) wird in einer Member-Variablen gespeichert.
- Die Member-Variablen dürfen wie üblich nicht von außen sichtbar sein, deine <<-Funktion soll aber schon direkt auf die Member zugreifen können.

Die Klasse soll folgende öffentliche Methoden bieten:

- Einen Standard-Konstruktor, der ein "leeres" Objekt (0 Messwerte) liefert. Für 0 Werte brauchst du kein dynamisch angelegtes Array: Setze den Array-Pointer auf den **nullptr**.
- Einen nur innerhalb der Klasse verwendbaren Konstruktor mit als Parameter angegebener Array-Größe: Er soll ein **Messw**-Objekt mit einem Array der angegebenen Größe liefern (die Array-Elemente selbst bleiben uninitialisiert).
- Einen Copy-Konstruktor und einen Destruktor mit der üblichen Funktionalität. Der Copy-Konstruktor ist außerhalb des **class** zu implementieren.

Verwende in allen Konstruktoren Initialisierungslisten!

- Einen unären ~-Operator, der die Anzahl der Messwerte im Objekt liefert, sowie einen unären !-Operator mit **bool**-Ergebnis, der **true** liefert, wenn das Objekt leer ist (0 Werte enthält).
- Einen []-Operator nur zum Lesen (nicht zum Ändern) einzelner Messwerte: Liegt der angegebene Index außerhalb des Arrays, soll er 0 als Ergebnis liefern, sonst den entsprechenden Messwert aus dem Array.
- Zwei +-Operatoren:
  - Einen +-Operator mit einem Messw-Objekt als linken und einem einzelnen double als rechten Operanden: Er soll ein Objekt liefern, in dessen Array zuerst die Werte des linken Operanden und dann der neue Wert gespeichert sind, also den Wert rechts an die bestehenden Werte anhängen.
  - Einen +-Operator mit zwei Messw-Objekten als Operanden: Er soll die Werte beider Objekte zusammenhängen, d.h. ein Objekt liefern, dessen Array zuerst die Werte des linken und dann die des rechten Operanden enthält.

**Hinweise** zu beiden Operatoren:

- Beide sind außerhalb des **class** zu implementieren.
- Du wirst den klasseninternen Konstruktor für das Ergebnis brauchen!

- Einen Typumwandlungs-Operator auf **double**:  
Er soll den Mittelwert aller Werte im Array liefern.  
Er ist außerhalb des **class** zu implementieren.  
Wenn das Objekt leer ist (das Array 0 belegte Werte enthält),  
soll 0 zurückgegeben werden.
- Einen Zuweisungsoperator mit der üblichen Funktion.  
Er ist außerhalb des **class** zu implementieren.  
Wenn das bestehende Array des Zielobjektes die falsche Größe hat,  
muss es in der richtigen Größe neu angelegt werden  
(Was machst du davor? Was musst du noch ändern?).  
Schütze dich auch gegen eine "Selbstzuweisung".

Weiters ist der Ausgabe-Operator << als Funktion zu implementieren.  
Er soll der Reihe nach alle Werte des Objektes ausgeben  
(mit Zwischenraum getrennt).

### Hauptprogramm:

Schreib zum Testen deiner Klasse ein Hauptprogramm.  
Es wird mit einem oder mehreren Dateinamen auf der Befehlszeile aufgerufen.

- Leg ein **Messw-Objekt** für alle Messwerte an ("Gesamt-Objekt").
- Geh in einer Schleife alle angegebenen Dateinamen der Reihe nach durch.  
Mach für jede einzelne Datei Folgendes:
  - Öffne die angegebene Datei zum Lesen.  
Prüfe, ob sich die Datei wirklich öffnen hat lassen,  
gib sonst eine Meldung aus (wohin?) und überspringe diese eine Datei  
(d.h. mach mit der nächsten Datei weiter)!
  - Leg ein **Messw-Objekt** für die Messwerte dieser Datei an.
  - Lies in einer weiteren Schleife immer wieder einen **double**-Wert aus der Datei  
und "addiere" ihn zum **Messw-Objekt**, bis das Lesen eines Wertes fehlschlägt.
  - Stelle mit dem Operator, der auf ein leeres Messw-Objekt prüft, fest,  
ob Werte gelesen wurden, und gib nach dem Lesen der Werte den Dateinamen  
und
    - entweder "**Keine Werte**" (wenn keine Werte gelesen wurden)
    - oder die aus dieser Datei gelesenen Werte aus.
 Gib danach Mittelwert und Anzahl der aus dieser Datei gelesenen Werte aus.
  - "Addiere" schließlich das Objekt mit den Werten dieser Datei  
zum "Gesamt-Objekt" dazu.
- Gib am Ende den ersten und den letzten Messwert aus dem Gesamt-Objekt  
sowie den Mittelwert und die Anzahl aller Werte im "Gesamt-Objekt" aus.

### Zusatzaufgabe:

Unser Hauptprogramm enthält an zwei Stellen "**xxx = xxx + yyy;**" für Messwerte.

Kannst Du in der **Messw**-Klasse zwei Operatoren += implementieren,  
die einen einzelnen **double** bzw. ein **Messw-Objekt** an ein **Messw-Objekt** anhängen,  
damit man im Hauptprogramm zweimal "**xxx += yyy;**" schreiben kann?