# Programmieren C++, Abgeleitete Klassen

## *Einfache Array-Klasse mit Statistiken*

Gesucht ist eine Klasse, die ein ganz einfaches Array implementiert,
und eine davon abgeleitete Klasse,
die zusätzlich die Schreibzugriffe auf das Array mitzählt.

Du kannst dazu von meiner Webseite einen Programm-Rahmen herunterladen,
der bereits **`main`** und eine Testfunktion enthält (die beiden sollst du *nicht* ändern!).
Auch das erwartete Ergebnis kannst du dir zum Vergleich auf meiner Webseite ansehen.

Die Basisklasse **`Array`** soll zwei *private* Membervariablen haben:

- Die *Größe* des Arrays.
- Einen *Pointer* auf das eigentliche Array.
  Es soll ein **`double`**-Array sein, das im Konstruktor
  mittels **`new`** in der gewünschten Größe dynamisch angelegt wird.
  (wo wird es wohl wieder freigegeben?)

Die **`Array`**-Klasse soll folgende öffentliche Methoden haben,
die "vorbeugend" alle *virtuell* sein sollen:

- *Keinen* Standard-Konstruktor.
- Einen *Konstruktor* mit einem **`int`**-Argument: Der gewünschten Größe des Arrays.
  (du musst den Inhalt des Arrays *nicht* initialisieren)
- Einen *Copy-Konstruktor*. (Was muss der wohl tun?)
- **`getSize()`**: Returniert die *Größe* des Arrays.
- **`getVal(`***index*`)`**: Liefert den *Wert* des Elementes Nummer *index*
  (*index* beginnt bei 0).
  Ist *index* außerhalb des Arrays, soll 0 zurückkommen.
- **`setVal(`***index*`, `*wert*`)`**: *Setzt* den Wert des Elementes Nummer *index*.
  Liefert ein boolsches Ergebnis: Wahr bei Erfolg, falsch bei ungültigem *index*.
- **`setAll(`***wert*`)`**: Setzt alle Elemente auf den angegebenen Wert. Kein Ergebnis.
- **`print()`**: Gibt alle Elemente aus
  (in einer Zeile mit Zwischenraum, abschließend ein **`endl`**).

***Hinweise*:**

- Da alle Methoden bis auf den ersten Konstruktor und **`getSize`** etwas länger sind,
  sollten sie *nicht* inline sein!
- Wenn du für alle **`int`**-Variablen, -Argumente und -Member "**`unsigned int`**"
  statt "**`int`**" als Typ verwendest, sparst du dir ein paar lästige Prüfungen,
  ob die Array-Größe negativ ist.
- Man darf mit **`new`** auch ein Array der Größe 0 anlegen
  (solange man auf das Ergebnis nicht zugreift).
  Das Freigeben eines solchen Arrays mit **`delete`** funktioniert ebenfalls.

  Diesen Fall muss man also nicht abfangen und extra behandeln.

Die abgeleitete Klasse **`StatArray`** hat *eine private Member-Variable*:

- Die *Anzahl der Schreibzugriffe* (**`setVal`**-Aufrufe, **`setAll`** wird nicht mitgezählt).

Weiters implementiert sie folgende *öffentliche Methoden*:

- Dieselben beiden *Konstruktoren* wie die Basisklasse.
  (die Anzahl der Schreibzugriffe wird beim Kopieren eines **`StatArray`**
  im neuen Objekt auf 0 gesetzt, nicht vom alten Objekt übernommen).
- **`setVal`** wird überschrieben:
  Hier muss mitgezählt und das **`setVal`** der Basisklasse aufgerufen werden.
- **`getWriCnt()`**: Liefert die Anzahl der bisherigen Schreibzugriffe.
- Alle anderen Methoden werden *unverändert* geerbt.