

Statistik-Objekt SimpleStat:

In C++ (aber auch in C) fasst man am besten alle Variablen, die zur Berechnung und Ausgabe der statistischen Kenngrößen benötigt werden, zu einem Objekt (`struct` oder `class`) zusammen. Ein Schreibzugriff des Users auf diese Variablen wäre fatal, sodass man sie als `private`: kennzeichnen sollte (das geht nur in C++). Die Interaktion des Anwenders mit dem Objekt beschränkt sich auf 3 Dinge (3 Objektmethoden):

- 1) Initialisierung (Konstruktor): Dieser muss alle Variablen richtig initialisieren (nullen!).
- 2) Einen neuen Wert `x` verarbeiten (ich finde `operator+=()` gut)
- 3) Ausgabe der berechneten Kenngrößen (`operator<<()`)

Wir brauchen im Statistikobjekt maximal:

```
Titel (Beschreibung der Daten): std::string title ;
Stichprobenzähler: int (oder unsigned) n;
Summenvariable (akkumulierte Summe der Werte): double sum;
Summenvariable (akkumulierte Summe der Quadrate): double sum2;
Variable für Minimum, Maximum, falls gewünscht: double min_val, max_val;
Speicher für die Werte: std::vector<double> data;
```

Da man den Datenspeicher nicht immer braucht (nur wenn der Median gewünscht wird), definiere ich 2 Objekttypen: `SimpleStat` (ohne Speicher), `StoringStat` (mit Speicher).

```
class SimpleStat {
public:
    explicit SimpleStat(std::string); // nur Konstruktor
    void operator+=(double); // eine Stichprobe verarbeiten
    void print(std::ostream&) const; // Ausgabe der Kennzahlen
private:
    std::string title; // Beschreibung der Daten
    int n{}; // Anzahl der Stichproben
    double sum{} // Summe der Daten, Initialisierung mit 0
    double sum2{} // Summe der Daten-Quadrate
    double min_val{}
           max_val{}; // optional, nur wenn man Extrema ermittelt
};
```

Ich habe hier für die Ausgabe eine eigene Objektmethode `print()` vorgesehen, weil das bei der Programmierung von `StoringStat` Vorteile hat. Aber natürlich braucht man auch die echte Streamausgabe für `SimpleStat`:

```
ostream& operator<<(ostream& os, const SimpleStat& s)
{    s.print(os); return os; }
```

Das neue Statistik-Objekt kann man wie einen eingebauten Typ verwenden:

```
SimpleStat daten{"Meine Daten"};
while (es gibt einen neuen Wert x)
    daten += x;
cout << daten;
```