

Die move-Varianten

Seit C++11 gibt es auch den Move-Konstruktor (analog zum Kopier-Konstruktor) und den Move-Zuweisungsoperator (analog zum Kopier-Zuweisungsoperator). Man setzt sie ein, um die Ressourcen des Quellobjekts zu stehlen und in das Zielobjekt zu verschieben (deshalb Move). Natürlich sollte man nur temporäre Quellobjekte beklaunen, da diese ohnehin gleich zerstört werden. Deshalb sind diese Argumente immer rvalue-Referenzen. Die Signaturen sind:

```
Objekttyp(Objekttyp&& temporary_original);  
Objekttyp& operator=(Objekttyp&& temporary_original);
```

Das Verschieben von Ressourcen kann viel schneller als das Kopieren sein, sodass diese Varianten sehr sinnvoll sein können. Alle STL-Containerklassen besitzen solche Methoden, ebenso die `std::string` Klasse. Ältere Programme, die diese Container benutzten, konnten durch Neukompilierung mit einem C++11 Compiler 30% und mehr beschleunigt werden (ohne eine Zeile Code im Programm zu ändern!).

C++ fügt in bestimmten Fällen auch diese 2 Methoden automatisch hinzu, allerdings sind die Bedingungen dafür viel restriktiver als bei den anderen speziellen Objektmethoden. Kurz zusammengefasst kann man sagen: Je weniger spezielle Objektmethoden vom Programmierer stammen, desto eher erzeugt C++ alle Move-Varianten.

Man kann C++ wieder zu seinem Glück zwingen:

```
Bruch(Bruch&&) = default; // erzwingt Erzeugung  
constexpr Bruch(Bruch&&) noexcept = default; // mit diesen Zusätzen  
Bruch& operator=(Bruch&&) = delete; // verhindert Erzeugung
```