

## Objektorientierung und Vererbung in C++

### Lernziele

Hier soll das Thema *Vererbung* vertieft und angewendet werden.

### Aufgabe 7:

Erweitern Sie in einer Klasse `ExtStack` die Funktionalität der Klasse `Stack` aus Aufgabe 3, indem Sie die folgenden Methoden implementieren:

- `size_t ExtStack::size()` liefert die Anzahl der aktuell im Stack gespeicherten Werte.
- `void ExtStack::swap(ExtStack& s)` tauscht die Elemente des Stacks mit den Elementen des Stacks `s` aus.

Der zusätzliche Konstruktor `ExtStack::ExtStack(const float *arr, size_t len)` soll die Elemente des als Parameter übergebenen Arrays `arr` in den Stack übernehmen.

Ändern Sie Ihren Testtreiber und erweitern Sie ggf. die Testfälle, um die Korrektheit Ihrer Implementierung nachzuweisen.

### Zusatz-Aufgabe Z7:

Erweitern Sie die Funktionalität der aus der Vorlesung bekannten Klasse `Rational` um eine Methode `double value()`, die den durch Zähler und Nenner repräsentierten Wert als `double` liefert. Die erweiterte Klasse soll `ExtRational` heißen.

Zahlen werden in den einzelnen Ländern unterschiedlich dargestellt. So werden in England die Tausender durch Kommata getrennt, in Deutschland durch Punkte:

1,234,567 England  
1.234.567 Deutschland

Andererseits werden in Deutschland die Nachkommastellen durch ein Komma vom ganzzahligen Teil getrennt, in England durch einen Punkt:

742.13 England  
742,13 Deutschland

Implementieren Sie eine Methode `std::string ExtRational::toString(int lk)`, die den Wert des Bruchs in der entsprechenden Länderkennung (lk) liefert. Dazu ist es sinnvoll, sich das Thema *locale*<sup>1</sup> einmal anzusehen.

Wie immer gehört das Erstellen von Testtreibern und Testfällen zur Aufgabe, um die Korrektheit Ihrer Implementierung nachzuweisen.

---

<sup>1</sup><https://en.cppreference.com/w/cpp/locale/locale>