

C++ für JAVA-Programmierer

Sommersemester 2025

Abgabe per Email bis Freitag, 15. August 2025

Möglichst in Gruppen (2 bis 4 Personen) bearbeiten !

naeher@uni-trier.de

Aufgabe 1 (10 Punkte)

Implementieren Sie eine C++ Klasse `array` zur Darstellung von Feldern von ganzen Zahlen. Neben den üblichen Methoden (Konstruktor, Zugriff auf Elemente, Ein/Ausgabe ...) soll die Klasse eine Funktion `void sort()` enthalten, die das Feld aufsteigend sortiert. Welchen Sortieralgorithmus Sie implementieren, ist Ihnen überlassen. Schreiben Sie ein C++ Programm, das Ihre Array-Klasse mit Zufallszahlen testet.

Aufgabe 2 (10 Punkte)

Schreiben Sie eine Klasse `fraction` zur Darstellung rationaler Zahlen (d.h. Brüche) durch jeweils zwei ganze Zahlen Zahlen (vom Typ `int`) `numerator` und `denominator`.

Die Klasse soll folgendn Funktionen (neben dem Konstruktor) bereitstellen:

- `void simplify` – Kürzt den Bruch
- `float decimal` – Gibt den Bruch als Dezimalzahl (bzw. `float`) zurück

Zusätzlich sollen die üblichen arithmischen Operationen durch das Überladen der folgenden Operatoren implementiert werden.

- `fraction operator+` – berechnet die Summe von zwei Brüchen
- `fraction operator-` – berechnet die Differenz von zwei Brüchen
- `fraction operator*` – berechnet das Produkt von zwei Brüchen
- `fraction operator/` – berechnet den Quotienten von zwei Brüchen
- `bool operator==` – überprüft zwei Brüche auf (wertmäßige) Gleichheit (d. h. $\frac{1}{2}$ und $\frac{3}{6}$ sind gleich)
- `ostream& operator<<` – gibt den Bruch aus, bspw. als $\frac{3}{5}$. Falls der Nenner 1 ist, so soll der Bruch als "normale ganze Zahl" ausgegeben werden, also 2 statt $\frac{2}{1}$.

Aufgabe 3 (10 Punkte)

Schreiben Sie eine Klasse `item`, welche eine ganze Zahl `value` und einen Verweis `next` (auf ein Element vom Typ `item`) speichert.

Verwenden Sie die Item-Klasse zur Implementierung einer Klasse `stack`, die folgende Funktionen (neben dem Konstruktor und Destruktor) bereit stellt:

- `void push(int x)` – fügt `x` oben auf dem Stack ein
- `int top()` – liefert das oberste Element des Stacks zurück
- `void pop()` – entfernt das oberste Element vom Stack
- `bool empty()` – testet, ob der Stack leer ist
- `int size()` – gibt die Anzahl der Elemente im Stack zurück
- `void print()` – gibt die Elemente des Stacks nacheinander aus