

6. Aufgabenblatt

Der Abgabetermin für Ihre Lösungen ist Montag, der **19. Juni 2017** um 23⁵⁵ Uhr. Die Bewertung erfolgt auf der Grundlage Ihres `git`-Repositories zu diesem Zeitpunkt. Stellen Sie sicher, dass Sie bis zu diesem Zeitpunkt alle Änderungen in Ihr Repository übertragen haben und vergessen Sie nicht das `makefile`!

Aufgabe 15

Erstellen Sie eine einfache Klasse namens `Vektor`, die einen dreidimensionalen numerischen Vektor speichern soll.

- ▶ Der Konstruktor erwartet drei Parameter vom Typ `double`, die den x -, y - und z -Komponenten des Vektors entsprechen.
- ▶ Werden die Parameter weggelassen, wird ein Nullvektor erzeugt.

Um mit den selbstdefinierten Vektoren arbeiten zu können, müssen noch spezifische Methoden implementiert und Operatoren überladen werden:

- ▶ Implementieren Sie eine Methode `betrag`, die den Absolutbetrag des Vektors berechnet.
- ▶ Überladen Sie die Operatoren `*` und `+` für das Kreuzprodukt und die Addition zweier Vektoren.
- ▶ Schreiben Sie eine Funktion `skalar` für die Berechnung des Skalarprodukts und eine Funktion `spat` für das Spatprodukt.

Für die Ausgabe eines Objekts dieser Vektorklasse mittels `cout` muss der `<<`-Operator überladen werden. Dabei sollen alle drei Komponenten des Vektors in eckigen Klammern ausgegeben werden, so dass

```
1 Vektor a(1,2,3); // Definition des Vektorobjekts
2 cout << a;      // Ausgabe
```

die Ausgabe `[1,2,3]` erzeugt.

- ▶ Recherchieren Sie, wie der `<<`-Operator überladen werden kann.
- ▶ Implementieren Sie die entsprechende Funktionalität.
- ▶ Kommentieren Sie die Überladungsfunktion für den `<<`-Operator besonders gründlich, indem Sie auf die Unterschiede zu den beiden anderen Überladungsfunktionen (für `*` und `+`) eingehen.

Das **Hauptprogramm** `aufgabe15.cpp` (mit der `main`-Funktion) zum Testen Ihrer Vektorklasse können Sie im Aufgabenbereich herunterladen. Es darf **nicht modifiziert** werden und sollte ohne Veränderungen mit Ihrer Implementierung der Vektorklasse funktionieren (siehe Aufgabenstellung).

(8 Punkte)

Beachten Sie bitte außerdem folgende Hinweise:

- ▶ Lagern Sie die Klassendefinition in eine Headerdatei namens `Vektor.hpp` aus.
- ▶ Fassen Sie den Konstruktor, die Methoden und die Überladungsfunktionen in einer Datei namens `Vektor.cpp` zusammen.
- ▶ Lesen Sie sich die Aufgabenstellung genau durch und fragen Sie, wenn etwas unverständlich ist.

Aufgabe 16

Schreiben Sie ein Programm, das eine „beliebige“ Anzahl von Fließkommazahlen aus einer Datei mit beliebigem Namen einliest und sortiert wieder ausgibt. Gehen Sie davon aus, dass jeweils eine Zahl in einer Zeile steht.

- ▶ Der Name der Datei mit den einzulesenden Zahlen soll beim Aufruf an der Kommandozeile an das Programm übergeben werden (siehe unten). Die Angabe des Dateinames ist nicht optional.
- ▶ Wenn das Programm ohne Parameter aufgerufen wird, sollte es eine kleine Hilfe zum korrekten Aufruf anzeigen und sich beenden.
- ▶ Fordern Sie den benötigten Speicher für die eingelesenen Daten dynamisch während der Laufzeit an.
- ▶ Recherchieren und implementieren Sie einen beliebigen Sortieralgorithmus (zum Beispiel *Bubblesort* oder *Quicksort*, siehe Internet und so) und geben Sie die sortierten Zahlen an der Standardausgabe aus.
- ▶ Hinweis: Für die Konvertierung von Zeichenketten in Fließkommazahlen ist die Funktion `atof` relativ gut geeignet.

(8 Punkte)

Beim Aufruf eines Programms werden die Anzahl der Argumente (als `int`-Wert) und ihr Inhalt (in Form eines Zeiger auf ein Feld von Zeichenketten) an die `main`-Funktion übergeben:

```
1  int main (int argc, char** argv)
2  {
3  ...  // Argumentauswertung
4  }
```

Die Anzahl der Parameter `argc` ist immer größer als Null, weil der Name des aufgerufenen Programms als erstes Argument in `argv[0]` enthalten ist.