

C/C++ Programmierung – Aufgabenblatt „Klassen 1“

Geschätzter Bearbeitungsaufwand außerhalb der Übungsstunden: 1 Stunde.

Schreiben Sie für die folgenden Klassen **jeweils eine Header- und eine .cpp-Datei!** Funktionen / Methoden **in der .cpp-Datei implementieren**. Verwenden Sie **Header-Guards** und **nicht #pragma once!**

1. Matrix

Schreiben Sie eine Klasse `Matrix`. Diese Klasse werden Sie im Laufe der folgenden Übungen erweitern. Die erste Version wird etwas seltsam sein – bitte nicht wundern, das wird später behoben (nötig dafür sind u.a. Arrays variabler Größe und die kommen erst in Kapitel 6).¹

- Legen Sie folgende Variablen als `protected Member` an:
 - je eine `int` Variable `m_Zeilen` und `m_Spalten`, die die Anzahl Zeilen bzw. Anzahl Spalten der Matrix enthalten sollen.
Vorläufig soll die Matrix die feste Größe 2x1 haben (also 2 Zeilen, eine Spalte).
 - ein Array `float m_Element[2]` (**exakt so** in der Klassendefinition hinschreiben!) für die 2*1 Elemente der Matrix. `m_Element` ist ein eindimensionales Array, in dem alle Elemente der Matrix hintereinanderstehen (zunächst die erste Zeile, dann die zweite usw. Das Array läuft also von 0 bis $< m_Zeilen * m_Spalten$).
Ein wie beschrieben angelegtes Array kann sofort verwendet werden: Ein `new` ist bei fester Größe *nicht* nötig.
- Legen Sie einen Standard-Konstruktor (Konstruktor ohne Parameter) an. Wie immer gilt: alle Variablen initialisieren! Verwenden Sie – sofern möglich – die Initialisierungsliste. Einzelne Elemente eines Arrays können *nicht* über die Initialisierungsliste gesetzt werden.
- Der Konstruktor soll zudem „Standardkonstruktor von Matrix wurde aufgerufen“ ausgeben.
- Schreiben Sie eine Methode `ausgabe()`, die alle Elemente der Matrix hintereinander ausgibt (unter Verwendung der Werte in `m_Zeilen` und `m_Spalten`).
- Schreiben Sie ein Hauptprogramm in einer *eigenen Datei*, das ein Objekt der Klasse anlegt und `ausgabe()` aufruft.

2. Vektor

Leiten Sie eine Klasse `Vektor` von der Klasse `Matrix` ab.

- Schreiben Sie eine Methode `betrag()`, die den Betrag des Vektors zurückgibt (so wie Sie es in MA2 gelernt haben).
Die Elemente des Vektors stehen in `m_Element[i]`.
Die Anzahl der Elemente ist `m_Zeilen * m_Spalten`.
- Die Klasse soll nur einen Standardkonstruktor haben und *keine* weiteren Methoden haben. Viel anfangen kann man mit dieser Klasse daher noch nicht – aber mit der Subklasse der nächsten Aufgabe!

3. Vektor2D

Leiten Sie eine Klasse `Vektor2D` von der Klasse `Vektor` ab.

- Schreiben Sie einen Konstruktor mit 2 Parametern für die 2 Elemente des Vektors und einen Standardkonstruktor.
- Schreiben Sie eine Methode `void addiere(Vektor2D v)`, die den 2D-Vektor `v` zu dem aktuellen Vektor addiert. Hierdurch wird der aktuelle Vektor also verändert!

4. Testprogramm zu Vektor2D

Erweitern Sie das Hauptprogramm, sodass es:

- drei Vektoren `a, b, c` vom Typ `Vektor2D` anlegt, wobei `a` mit (3,1) und `b` mit (1,2) initialisiert werden soll, während für `c` der Standardkonstruktor verwendet werden soll, und
- dann zu dem Vektor `c` den Vektor `a` addiert (hierdurch ändert sich `c`) und
- dann zu dem Vektor `b` den Vektor `c` addiert (hierdurch ändert sich `b`) und
- den Betrag der Summe berechnet und ausgibt.