

Übungsblatt 2: Einführung C++ (Update 1)

Abgabe:

Dieses Übungsblatt ist einzeln zu lösen. Die Lösung ist bis **Dienstag, den 11. Mai 2010, 12:00 Uhr s.t.** über UniWorx (<http://www.pst.ifi.lmu.de/uniworx>) abzugeben.

Es werden nur die Formate PDF und Plain-Text (UTF-8) akzeptiert. Erstellen Sie für jede Aufgabe ein Unterverzeichnis nach dem Schema <Übungsblatt>-<Aufgabe>, d.h. die Lösung der ersten Aufgabe kommt in ein Verzeichnis 2-1/. Packen Sie alle Dateien in eine ZIP-Datei und laden Sie diese bei UniWorx hoch. Wenn Sie Formatierungsvorgaben nicht einhalten, werden bis zu zwei Punkte abgezogen. Lösungen müssen zumindest im CIP-Pool fehlerfrei kompilieren und laufen. Bitte geben Sie nur Quellcode ab, keine kompilierten Dateien.

Inhalt:

Ziel dieses Übungsblattes ist, ein erste Erfahrungen mit C++ zu sammeln und das Wissen aus dem ersten Übungsblatt anzuwenden. Es können maximal 20 Punkte erreicht werden.

Vorbereitungen:

Laden Sie den QT Creator von <http://qt.nokia.com/downloads> herunter und installieren Sie ihn. Der QT Creator ist eine IDE für C++ mit guter QT-Unterstützung. Wir werden ihn im weiteren Übungsbetrieb verwenden. Stellen Sie sicher, dass ein C++-Compiler installiert ist (unter Windows wird die mingw-Toolchain mitinstalliert). Machen Sie sich mit der IDE vertraut.

Aufgabe 1: Pointer (5 Punkte)

Gegeben seien die folgenden Definitionen:

<pre>int x = 10; int y = 20; int z = 0; int* a = &x; int* b = &y; int** c = &a; // A y = x; x = 20; // B b = a; z = (int)&x; a = (int*)z; y = a* *a; // C</pre>	<p>Geben Sie für jeden der Zustände A, B, C an, welchen Wert a,b,c,d,x,y,z haben.</p> <p>Bei Pointern geben Sie an, welchen Wert sie erhalten würden, wenn sie alle Pointer auf dem Weg dereferenzieren würden.</p> <p>Beispiel im Zustand A: x = 10 c (c->a->x) = 10</p> <p>Beim Zustand C können Sie nicht für alle Variablen einen konkreten Wert angeben. Erklären Sie, weshalb.</p>
---	--

Geben Sie eine Datei 2-1/pointer.txt oder 2-1/pointer.pdf ab.

Aufgabe 2: Primitive, C++-Klassen (5 Punkte)

Laden Sie das Code-Gerüst für diese Aufgabe von der Website herunter. Es enthält eine Klasse CPrimitive, die ein geometrisches Primitiv repräsentiert, sowie eine Klasse CBox die von CPrimitive erbt. Die Datei main.cpp enthält lediglich die main()-Funktion.

- Implementieren Sie die in den Header-Dateien angegebenen Methoden. Der Mittelpunkt des Primitivs befindet sich am Punkt (0,0,0).
- Implementieren Sie analog zu CBox eine zusätzliche Klasse CPyramid, die ebenfalls von CPrimitive erbt. Der Konstruktor soll die Höhe und eine Seitenlänge der quadratischen Grundfläche der Pyramide erhalten.
- Die main()-Funktion soll nacheinander eine Box und eine Pyramide erstellen und deren Daten auf der Standardausgabe drucken.

Legen Sie alle Quellcode-Dateien in einem Verzeichnis "2-2" an.

Aufgabe 3: 3D-Transformationen (10 Punkte)

Implementieren Sie ein Programm um 3D-Transformationen auf Vektoren anzuwenden.

- Erzeugen Sie alle Dateien im Verzeichnis 2-3.
- Implementieren Sie eine Klasse CTransform mit den folgenden Eigenschaften:
 - Der Konstruktor erhält keine Parameter.
 - Intern wird eine Transformationsmatrix durch ein zweidimensionales Array *mat* repräsentiert. Stellen Sie sicher, dass nur die Klasse selbst Zugriff darauf hat. Bei der Initialisierung des Objekts soll diese Matrix mit der Einheitsmatrix initialisiert werden.
 - die public-Methoden `rotate(Axis a, float degrees)`, `translate(Axis a, float distance)` und `scale(Axis a, float factor)` modifizieren die Transformationsmatrix *mat*. Implementieren Sie dies dergestalt, dass sie ein temporäres Array mit den entsprechenden Werten füllen und dieses mit der Transformationsmatrix *mat* multiplizieren. **Tipp: Implementieren Sie Axis als enum.**
 - die public-Methode `transform(float* vector)` erhält einen Vektor als eindimensionales float-Array und gibt einen transformierten Vektor vom gleichen Typ zurück.
- Die main()-Funktion soll von der Standardeingabe drei Integer-Werte lesen, die durch Leerzeichen getrennt sind (*cin*), und nach einem Druck auf die Return-Taste aus diesen einen Vektor erzeugen. Dieser soll von einem CTransform-Objekt um 10 in X-Richtung verschoben, um -90° um die Z-Achse rotiert und um den Faktor 3 in Y-Achse skaliert werden. Der resultierende Vektor soll auf der Standardausgabe ausgegeben werden. Anschließend soll wieder ein neuer Vektor eingelesen werden, usw. Sie können voraussetzen, dass nur gültige Integer-Werte eingegeben werden.
- mit `make all` soll das ganze Programm kompiliert werden, mit `make clean` sollen alle kompilierten Dateien gelöscht werden..

Viel Erfolg.