UNIVERSITÄT OSNABRÜCK

Dr. Thomas Wiemann

8. Übungsblatt zur Vorlesung „Einführung in die Programmiersprache C++“

Wintersemester 2022 / 2023

Aufgabe 1: Implementierung einer generischen Liste (30 Punkte)

In dieser Aufgabe implementieren Sie eine generische einfach-verkettete Liste mittels C++-Templates. Ihre Liste soll mindestens die folgenden Methoden zur Verfügung stellen

```
void insert(T item);

void for_each(void (*do_something)(T item));
```

Durch Aufruf der `insert`-Methode soll ein neues Element des generischen Typs `T` in die Liste eingefügt werden. Durch Aufruf der `for_each`-Methode soll über alle in der Liste gespeicherten Elemente iteriert werden. Dabei wird für jeden Knoten der Liste die durch den Parameter `do_something` spezifizierte Funktion mit dem Inhalt des Knotens aufgerufen. Testen Sie Ihre Listenimplementierung zunächst mit Integer-Werten, indem Sie 10 Zahlen in die Liste einfügen. Testen Sie `for_each` mittels einer geeigneten Funktion in einem kleinen Testprogramm, das die zuvor gespeicherten Zahlen auf der Konsole ausgibt.

Aufgabe 2: Integration der Liste in die Klasse MainWindow (20 Punkte)

In der bisherigen Struktur werden die `render()`-Methoden der zu zeichnenden Objekte durch separate Aufrufe in der Methode `execute()` der Klasse `MainWindow` aufgerufen. Integrieren Sie die in Aufgabe 1 implementierte Liste in das vorgegebene Programmgerüst, indem Sie Zeiger auf alle zu rendernden Objekte zunächst in einer geeigneten Liste speichern. Benutzen sie anschließend die `for_each`-Methode der Liste, um die Objekte zu rendern. Hinweis: Fügen sie der Klasse `MainWindow` eine geeignete statische `render`-Methode hinzu, die die `render()`-Methode eines `Renderables` aufruft. Übergeben sie diese statische Methode dann als Parameter an `for_each`.

Aufgabe 3: Umwandlung der Vector-Klasse in ein Template (50 Punkte)

In unserem Programm benutzen wir die Klasse `Vector` um 3D-Koordinaten zu repräsentieren. In diesem Teil der Übung werden wir die Klasse generalisieren. D.h., sowohl der Typ der Koordinaten als auch die Länge des Vektors sollen durch Templates repräsentiert werden. Dies wird uns ermöglichen, geeignete Spezialisierungen sowohl für 2D-Bildschirmkoordinaten in Integer-Repräsentation als auch für 3D-Weltkoordinaten mit Floating-Point-Werten zu verwenden. Ersetzen Sie dazu die im Repository vorhandenen Klassen `Vector3f` und `Vector2i` durch eine generische Klasse `Vector`. Diese soll durch geeignete `typedef`-Definitionen die Klassen `Vector3f` und `Vector2i` ersetzen. Gehen Sie dazu wie folgt vor:

1. Implementieren Sie die generische Klasse `Vector` mit allen vorhandenen Operatoren und Funktionalitäten der Klasse `Vector3f`. Fügen Sie dazu die Template-Parameter `T` für den Typ der Koordinaten und den Parameter `L` für die Länge des Vektors ein. Fügen Sie der Klasse ein statisches Array der Länge `L` hinzu und implementieren Sie einen Konstruktor mit drei Parametern `x`, `y` und `z`, der die Elemente dieses Arrays initialisiert. Bei 2D-Vektoren soll der dritte Parameter ignoriert werden. Die Werte dieser Parameter sollen per Default auf `0` gesetzt werden.
2. Stellen Sie sicher mittels eines `static_assert`-Checks sicher, dass der Parameter `L` im gültigen Wertebereich (also 3 oder 2) ist. Welchen Vorteil hat die Verwendung eines solchen Assert-Checks?
3. Checken Sie in den Index-Operatoren den Wert des übergebenen Indexes auf einen gültigen Wert. Werfen Sie eine `std::invalid_argument`-Exception, falls dieser sich außerhalb des gültigen Wertebereichs befindet. Warum ist ein Assert analog zu Punkt 2 in diesem Fall nicht sinnvoll?

4. Wenn Sie alle geforderten Funktionalitäten implementiert haben, Fügen sie dem Header `Vector.hpp` eine passende `typedef`-Deklaration hinzu, die die Spezialisierungen `Vector3f` für Weltkoordinaten und `Vector2i` für Bildschirmkoordinaten definiert.

Trennen Sie bei der Implementierung die Deklaration von der Implementierung. Lagern Sie die Implementierungen geeignet in eine `.tcc`-Datei aus.

Abgabe:

Checken Sie Ihre Lösung des Aufgabenblattes bis Montag den 19.12.2022, 8:00 Uhr in den Master-Branch des git-Repositorys Ihrer Gruppe ein.

https://xkcd.com/379/