

# 18.911: Kompaktkurs C++ für Java-Programmierer

Reinhard Zierke

Universität Hamburg, MIN-Fakultät, Informatik-RZ

8.-12. Oktober 2007

<http://www.informatik.uni-hamburg.de/RZ/lehre/C++/>

## Aufgabe 4.1

[Lernziel: Templates]

Reimplementieren Sie die  $4 \times 4$ -Matrix-Klasse aus Aufgabe 3.1 als Template, so dass sie beliebige passende Elemente (`int`, `double`, `complex`) speichern kann.

## Aufgabe 4.2

[Lernziel: Implementation einer Klasse mit Speicherverwaltung]

Erweitern Sie die Matrix-Klasse aus Aufgabe 3.1/4.1 auf beliebige Größe, indem Sie Heap-Speicher nutzen. Stellen Sie den Speicher im Konstruktor bereit und geben ihn im Destruktor frei.

Hinweis: Die Aufgabe ist leichter zu lösen, wenn Sie einen zusammenhängenden Speicherbereich von  $nrows \times ncols$  Elementen benutzen und die Indizierung mit überladenem `()`-Operator programmieren ( $m(y, x) = m.daten[y \times ncols + x]$ ).

## Aufgabe 4.3

[Lernziel: Iteratoren]

Reimplementieren Sie die ASCII-Bild-Aufgabe 2.3, indem Sie zur Bildspeicherung Vektoren von Vektoren sowie zum Zugriff auf die Bildpunkte Iteratoren statt Arrayindizierung benutzen.

## Aufgabe 4.4

[Lernziel: STL-Datenstrukturen und Algorithmen]

Bestimmen Sie die Worthäufigkeit in einem Text: Lesen Sie eine Textdatei (Beispiele auf der Kurswebseite) wortweise ein und zählen Sie die Häufigkeit jedes Wortes in ei-

ner indizierten Tabelle vom Typ `map<string,int>`. Geben Sie diese Worttabelle nach Häufigkeit sortiert aus.

Hinweis: Auf ein einzelnes Element Ihrer Tabelle können Sie mit dem `[]`-Operator zugreifen: `worte[str] = 1;`

Hinweis: Intern sind die Elemente einer `map` vom Typ `pair`. Die beiden Elemente eines solchen Paares können Sie mit den Datenfeldnamen `first` und `second` ansprechen. Sie also mit einem `map`-Iterator `iter` über Ihre indizierte Tabelle laufen, können Sie mit `iter->first` den aktuellen Index und mit `iter->second` den zugehörigen Wert ansprechen.

Hinweis: Zur sortierten Ausgabe können Sie ihre Tabelle in eine neue Tabelle vom Typ `multimap<int,string>` umkopieren, in der die Häufigkeit der Index und das Wort der Wert ist. Eine `multimap` kann anders als eine `map` mehrere Werte (hier Wörter) für einen Index aufnehmen. Wenn Sie diese `multimap` rückwärts ausgeben, haben Sie damit eine nach Häufigkeiten absteigende sortierte Liste.

## Aufgabe 4.5

[Lernziel: STL-Datenstrukturen und Algorithmen]

Erweitern Sie die polymorphe Version der Klasse `Person` (siehe Website) wie folgt:

1. Ergänzen Sie die Klasse `Person` um eine weitere Komponente, ein dynamisches Objekt vom Typ `Adresse`. Dieses ist eine neue Klasse mit den Komponenten

PLZ Ort  
Straße Hausnummer

Da die Adresse eine dynamische Komponente ist (Datenfeld als Referenz; Objekt erzeugen und löschen mit `new` bzw. `delete`), müssen Sie auch den Kopierkonstruktor und Zuweisungsoperator neu schreiben.

2. Wählen Sie einen geeigneten Containertyp aus der STL und erstellen Sie eine Klasse `Database`, die Personen als Elemente aufnehmen kann. Fügen Sie mehrere Personen in Ihre Datenbank ein und erzeugen Sie eine nach Nach- und Vornamen alphabetisch sortierte Liste. Dabei sollten sie den Algorithmus `sort` der STL benutzen.

Hinweis: Damit Sie mit `sort` sortieren können, müssen Sie in der `Person`-Klasse zwei boolsche Operatoren `==` und `<` definieren:

`==` ist wahr, wenn der Name der beiden Personen gleich ist.

`<` ist wahr, wenn der Nachname meines Objektes alphabetisch kleiner als der des Arguments ist, oder wenn bei gleichem Nachnamen der Vorname kleiner ist.