



Lehrveranstaltung für TI-Bachelor: "Vertiefung objektorientierte Programmierung in C++" Übungsblatt 7

Hinweise:

Dieses Übungsblatt ist zur Zulassung zu der Klausur erfolgreich zu bearbeiten ("*Erfolgreich*" bedeutet: Keine Programmabstürze bzw. Endlosschleifen, Aufgabenstellung einschl. der Nebenbedingungen müssen eingehalten sowie Kommentierung und Einrückung korrekt sein!).

Die Aufgaben werden überwiegend in den Übungszeiten bearbeitet. Allerdings genügt die Zeit hierfür unter Umständen nicht, so dass Sie auch außerhalb dieser Zeiten die Aufgaben bearbeiten müssen. Der Abgabetermin für diese Aufgabe ist **spätestens** der **30. Januar 2014**.

Nutzen Sie die Übungen auch, um ggf. Fragen, die sich in den Vorlesungen ergeben haben, anzusprechen.

Aufgabe: In der siebenten und letzten Übungsaufgabe des Projektes „Belegungen“ soll in der Klasse `TBookings` die Datenstruktur der Belegungen geändert werden. Die Belegungen sollen (bisher in einem Vektor) jetzt in einer doppelt verkettete Liste abgelegt werden. Die doppelt verkettete Liste soll eine eigene Template-Klasse sein und zusätzlich Iteratoren enthalten (siehe Kapitel 14.3). Dazu dürfen Sie die vorgegebene Headerdatei `tliste.h` verwenden.

Die Klasse `TBookings` muss entsprechend umgestellt werden; vor allem sollen natürlich die Iteratoren verwendet werden!

Auch sollen in dieser Klasse zwei Suchfunktionen zur Suche von Belegungen von einem Studenten implementiert werden:

`findFirstBookingFromStudent` (Parameter Name des Studenten)
und

`findNextBookingFromStudent` (ohne Parameter).

Beide Funktionen geben jeweils einen Zeiger auf eine Belegung (`TBooking`) zurück, wenn eine Belegung gefunden wurde, bzw. einen `NULL`-Zeiger, wenn keine (weitere) Belegung gefunden wurde.

Ferner soll die letzte Klasse implementiert werden: `TSchedule`. In ihr soll ein Titel (`string`) und ein zweidimensionales Array von Zeigern auf Belegungen eingefügt werden, das dann mit Hilfe des Ausgabe-Operators als Stundenplan (siehe Beispielausgabe) ausgegeben werden soll. Eine Methode (z.B. `getStudentsplan`) soll die passenden Belegungen von dem angegebenen Studenten aus der Liste der Belegungen heraus-suchen und in das zweidimensionale Array eintragen.

Im Skript ist im Beispiel für die doppelt verkettete Liste mit Iteratoren ein kleiner Fehler: Bei der `friend`-Deklaration muss es heißen

`friend class Liste<T>;`

statt

`friend Liste<T>;` bzw. `friend Liste;`

In der vorgegebenen Headerdatei ist dieser Fehler natürlich nicht enthalten.

Zum Testen der Klassen soll wieder ein kleines Hauptprogramm geschrieben bzw. kann das vorgegebene Beispielprogramm verwendet werden, dass Objekte der angegebenen Klassen erzeugt, auf verschiedene Werte setzt und wieder auf dem Bildschirm ausgibt.

Beispiel für ein Hauptprogramm:

```
#include <iostream>
#include <cstdio>

using namespace std;

#include "tbookings.h"
#include "tschedule.h"

int main()
{
    TBookings Belegung("data.xml");
    TSchedule Stdplan;

    Belegung.print();
    cout << endl;

    Stdplan.getStudentsplan(Belegung, "Detlef Dicht");
    cout << Stdplan << endl;

    return 0;
}
```

Beispiel-Ausgabe:

Belegungen Studenten:

=====

Anna Albrecht:

- 3: Informatik I (WS2013/14) (belegt am 15.09.2013 um 01:28; Rang 2)
 4: Informatik I Ubg (WS2013/14) (belegt am 15.09.2013 um 01:29; Rang 2)

Detlef Dicht:

- 5: Informatik I (WS2013/14) (belegt am 16.09.2013 um 09:51; Rang 3)
 6: Informatik I Ubg (WS2013/14) (belegt am 16.09.2013 um 09:55; Rang 3)
 8: Mathe I (WS2013/14) (belegt am 19.09.2013 um 22:11; Rang 2)

Frank Faust:

- 1: Informatik I (WS2013/14) (belegt am 15.09.2013 um 00:14; Rang 1)
 2: Informatik I Ubg (WS2013/14) (belegt am 15.09.2013 um 00:17; Rang 1)
 7: Mathe I (WS2013/14) (belegt am 17.09.2013 um 17:48; Rang 1)

Max Muster:

Susi Sorglos:

Stundenplan von Detlef Dicht

	Mo	Di	Mi	Do	Fr	Sa
1						
2						
3					Informatik I Kevin Kaufmann D 419	
4	Mathe I Nathan Neuling H5 (D136)				Informatik I Ubg Kevin Kaufmann DE16b	
5						
6						
7						
8						