



## Lehrveranstaltung für TI-Bachelor: "Vertiefung objektorientierte Programmierung in C++" Übungsblatt 3

### 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 **12. Mai 2016**.

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

**Aufgabe:** In der dritten Übungsaufgabe des Projektes "Turniermanager" sollen die Klassen `TStadium` und `TTournament` erstellt werden. Ferner sollen alle Daten aus einer vorgegebenen Datei im XML-Format eingelesen werden.

Die Klasse `TStadium` erhält als private Eigenschaft den Namen des Stadiums (`string`), die Stadt und das Land (`Town` und `Country`; jeweils `string`), in dem das Stadium steht, sowie die Anzahl der Sitzplätze (`NumberOfSeats`; `int`) und das Baujahr (`YearOfConstruction`; `int`). Natürlich müssen auch noch Konstruktoren, `set`-, `get`- und `print`-Methoden erstellt werden (siehe Beispielausgabe).

Die Klasse `TTournament` erhält als private Eigenschaft den Namen des Turniers (`string`), die Anzahl der eingetragenen Mannschaften (`NumberOfTeams`; `int`), ein Array mit den Mannschaften (Array von `TTeam`; die Größe des Arrays wird mit einer Konstanten `MAXTEAMS` auf 12 festgelegt), sowie die Anzahl der eingetragenen Stadien (`NumberOfStadiums`; `int`) und ein Array mit den Stadien (Array von `TStadium`); die Größe des Arrays wird mit einer Konstanten `MAXSTADIUMS` auch auf 12 festgelegt). Die Anzahl der Spiele (`NumberOfMatches`; `int`) und das Array mit den Spielen wird erst in der nächsten Übungsaufgabe nachgetragen. Der Konstruktor dieser Klasse erhält nur einen Dateinamen. Diese Datei soll im Konstruktor geöffnet werden. Anschließend sollen die Daten von den jeweiligen Klasse eingelesen werden. Natürlich muss am Ende des Konstruktors die Datei wieder geschlossen werden!

Zum Einlesen der Daten sollen alle Klassen (TDate, TPlayer, TTeam und TStadium; die anderen Klassen kommen beim nächsten Mal dran) jeweils um eine Methode namens `load` erweitert werden; als Parameter erhalten sie eine Referenz auf einen Datei-Eingabedatenstrom (`ifstream`) – geöffnet wird die Datei ja im Konstruktor der Klasse `TTournament`. In den `load`-Methoden wird zeilenweise aus der Datei gelesen und dann entsprechend des gelesenen XML-Tags der Inhalt in die Eigenschaft der Klasse übernommen oder die `load`-Methode der entsprechenden Klasse aufgerufen.

Z.B. wird beim Einlesen des Turniers das Tag `<Team>` gelesen. Dann wird die `load`-Methode der Klasse `TTeam` aufgerufen. Dort werden dann Name, Trainer, usw. des Teams eingelesen. Es wird aber auch das Tag `<Player>` kommen; dann muss mit `new` ein neuer Spieler erzeugt werden und anschließend die `load`-Methode des neuen Spielers aufgerufen werden. Dieser liest die Daten eines Spielers ein, bis das Tag `</Player>` erscheint, und springt anschließend in die `load`-Methode des Teams zurück, in der dann weitere Daten gelesen werden können (bis dort das Tag `</Team>` auftaucht). So soll jede Klasse nur die eigenen Eigenschaften einlesen und bei anderen Eigenschaften die `load`-Methode der entsprechenden Klasse aufrufen.

Hinweise:

- Es können beliebig viele Teams, Spieler und Stadien in der Datei stehen; in den nächsten Übungen stehen hier auch noch die Daten der einzelnen Spiele und deren Events.
- Es müssen nicht immer alle Tags existieren. Wenn ein Spieler z.B. keine roten Karten hat, kann der Tag auch fehlen.
- Die Reihenfolge der Tags kann variieren!
- Die Anzahl der Leerzeichen am Zeilenanfang kann variieren!

Zum Testen der Klasse soll wieder das vorgegebene Hauptprogramm verwendet werden, das ein Turnier-Objekt (mit Angabe der einzulesenen Datei) erzeugt und anschließend die Daten auf dem Bildschirm ausgibt. Dabei liefern die Methoden `getTeam` und `getPlayer` jeweils einen Zeiger auf die jeweilige Mannschaft bzw. auf den jeweiligen Spieler.

## Hauptprogramm:

```
#include <iostream>

using namespace std;

#include "ttournament.h"

int main()
{
    TTournament Bundesliga("fussball.xml");

    Bundesliga.print();
    cout << endl;

    cout << "Erster und letzter Spieler in der Liste:" << endl;
    ((Bundesliga.getTeam(0))->getPlayer(0))->print();
    cout << endl;
    ((Bundesliga.getTeam(0))->getPlayer(
        (Bundesliga.getTeam(0))->getNumberOfPlayers() - 1))->print();
    cout << endl;

    return 0;
}
```

**Beispielausgabe:**

Turnier: 1. Bundesliga

Stadien:

- Olympiastadion (74475 Sitzplaetze) in Berlin; Deutschland
- Allianz Arena (75000 Sitzplaetze) in Muenchen; Deutschland

Teams:

Hertha BSC

Trainer: Pal Dardai

Spieler	Nr	Spiele	Tore	Vorl.	Gelb	Rot	Geburtstag
Rune Jarstein	22	25	0	0	1	0	29.09.1984
Thomas Kraft	1	5	0	0	0	0	22.07.1988
Marvin Plattenhardt	21	29	2	5	4	0	26.01.1992
Mitchell-Elijah Weiser	20	25	2	4	5	0	21.04.1994
John Brooks	25	20	1	1	3	0	28.01.1993
Sebastian Langkamp	15	19	0	1	7	0	15.01.1988
Per Skjelbred	3	28	0	1	8	0	16.06.1987
Vladimir Darida	6	28	4	3	6	0	08.08.1990
Genki Haraguchi	24	28	4	2	4	0	09.05.1991
Fabian Lustenberger	28	25	1	0	3	0	02.05.1988
Alexander Baumjohann	9	21	1	2	0	0	23.01.1987
Valentin Stocker	14	19	1	0	2	0	12.04.1989
Salomon Kalou	11	27	14	2	2	0	05.08.1985
Vedad Ibisevic	19	22	9	3	3	1	06.08.1984

Erster und letzter Spieler in der Liste:

Rune Jarstein (Trikot-Nr. 22)

Vedad Ibisevic (Trikot-Nr. 19)