

---

## Lehrveranstaltung "Algorithmen und Datenstrukturen" Übungsblatt 4

### Hinweise:

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

Die Aufgaben werden überwiegend in den Übungszeiten bearbeitet und dort auch abgegeben. 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 der **05. Juni 2026**.

---

**Aufgabe:** In der vierten Übungsaufgabe des Projektes "Mannschafts-Verwaltung" sollen die Daten beim Beenden des Programms in einer Datei gespeichert werden, damit sie beim nächsten Programmstart wieder eingelesen werden können.

Für das Laden und Speichern der Daten sollen einige Funktionen in einem neuen Modul `database.c` erstellt werden. Erzeugen Sie das Modul (C- und Headerdatei) und passen Sie das Projekt an.

Das Speichern der Daten soll im neuen Modul die Funktion `save` übernehmen. Hier soll nach dem erfolgreichen Öffnen der Datei die erste Zeile mit der Startkennung der Daten (`<Data>`; siehe Beispiel) geschrieben werden. Dann werden in einer Schleife die Daten der einzelnen Mannschaften geschrieben; das Schreiben einer Mannschaft kann eine Funktion namens `saveTeam` übernehmen (diese Funktion erhält als Parameter den Datenstrom der offenen Datei). Nach der Startkennung einer Mannschaft (`<Team>`) werden die Daten der Mannschaft (Mannschaftsname und Trainername) jeweils mit Start- und Endkennung in eine Zeile geschrieben (z.B. `<Name>Hertha BSC</Name>`). Anschließend werden die Spieler in einer Schleife gespeichert, z.B. mit einer Funktion `savePlayer`. Nachdem alle Spieler geschrieben wurden, wird die Endkennung der Mannschaft (`</Team>`) in eine eigene Zeile geschrieben; nach der letzten Mannschaft noch die Endkennung der Daten (`</Data>`).

Für das Laden der Daten soll eine Funktion `load` geschrieben werden. In dieser Funktion sollen nach dem erfolgreichen Öffnen der Datei die Daten eingelesen werden. Immer wenn die Startkennung einer Mannschaft gefunden wird, soll die Funktion `loadTeam` aufgerufen werden. Diese liest die Daten der einen Mannschaft ein – also bis zur Endkennung `</Team>`. Wird in dieser Funktion `loadTeam` die Startkennung eines Spielers gefunden, so soll die Funktion `loadPlayer` aufgerufen wer-

den, die die Daten eines Spielers einliest – also bis zur Endkennung `</Player>`.

Durch die Start- und Endkennungen können die Felder in beliebiger Reihenfolge stehen. Ferner kann darüber beim Einlesen geprüft werden, ob die Struktur der Daten gültig ist. Um die Datenbank kontrollieren zu können, sollte auf eine Verschlüsselung der Daten verzichtet werden. Überlegen Sie sich, welche Maßnahmen noch nötig sind, um angemessen auf eine fehlerhafte Datenbank zu reagieren. Was muss getan werden, wenn in einem Datensatz ein Feld fehlt (z.B. fehlen in der Beispieldatenbank mehrere Felder), damit dieser Datensatz trotzdem noch korrekt eingelesen und angezeigt werden kann.

Wem diese ganze Beschreibung zu kompliziert ist, guckt sich am besten erst das unten stehende Beispiel an; dies sollte vieles erklären.

Noch ein paar Hinweise:

1. Nach dem Laden und Speichern sollte die Datei natürlich jeweils wieder geschlossen werden.
2. Das Laden der Daten erfolgt bei Programmstart; das Speichern (optional mit Benutzerabfrage) bei Programmende. Wer möchte, kann alternativ Laden und Speichern als Menüpunkte einbauen (das Laden sollte dann die Daten zu den bereits eingegebenen Daten hinzufügen!).
3. Vor dem Einlesen eines jeden Datensatzes sollten die Felder mit Nullen (0 bzw. NULL) gefüllt werden.
4. Bei Programmende müssen natürlich (sofern nicht bereits in der vorigen Übungsaufgabe erledigt) alle reservierten Speicherbereiche wieder freigegeben werden!
5. Beim Einlesen kann immer eine ganze Zeile eingelesen werden. Um herauszufinden, welche Daten in der Zeile stehen (dieser Vorgang wird „parsen“ genannt), kann z.B. die Funktion `strncmp` aus der `string.h` verwendet werden; z.B. wird mit `if (strncmp(Zeile, "<Date>", 6) == 0)` geprüft, ob die ersten 6 Zeichen in der Zeichenkette Zeile gleich `<Date>` sind.
6. Am Anfang einer Zeile können beliebig viele Leerzeichen oder Tabulatoren stehen.
7. Innerhalb eines Datensatzes können Tags mehrfach vorkommen; entsprechend müssen die Speicherbereiche der bereits eingelesenen Daten wieder freigegeben werden.
8. Ihr Programm muss die mitgelieferte Beispieldatei einlesen können.
9. Wenn Sie Max Mustermann in der Bildschirmausgabe sehen, ist etwas schief gegangen.

**Generell soll immer mit Zeigern anstelle von Arrays gearbeitet werden! Oder anders ausgedrückt: Das Verwenden von eckigen Klammern ist tabu (außer bei der Definition von Arrays).**

Vergessen Sie nicht, dass bei Programmende **alle** reservierten Speicherbereiche wieder freigegeben werden müssen!

Kommentieren Sie das Programm. Dazu gehört auch ein Modulheader und zu jeder Funktion ein Funktionsheader (siehe Skript „Grundlagen der Informatik“ Kapitel 5.3 und 5.4)! Achten Sie auch auf Ihre Programmstruktur (Einrückungen, Leerzeichen und -zeilen).

**Auszug der Beispieldatei:**

```

<Data>
  <Team>
    <Player>
      <Name>Frederik Roennow</Name>
      <Birthday>4.8.1992</Birthday>
      <TricotNr>1</TricotNr>
    </Player>
    <Name>1. FC Union Berlin</Name>
    <Player>
      <Name>Matheo Raab</Name>
      <Birthday>18.12.1998</Birthday>
      <TricotNr>31</TricotNr>
      <Goals>0</Goals>
    </Player>
    <Trainer>Steffen Baumgart</Trainer>
    <Player>
      <Name>Danilho Doekhi</Name>
      <Birthday>30.06.1998</Birthday>
      <TricotNr>5</TricotNr>
      <Goals>0</Goals>
    </Player>
    <Player>
      <Name>Diogo Leite</Name>
      <Birthday>23.1.1999</Birthday>
      <TricotNr>4</TricotNr>
      <Goals>0</Goals>
    </Player>
    ...
    <Player>
      <Name>Marten Winkler</Name>
      <Birthday>31.10.2002</Birthday>
      <TricotNr>22</TricotNr>
      <Goals>0</Goals>
    </Player>
    <Player>
      <Name>Sebastian Groenning</Name>
      <Birthday>03.02.1997</Birthday>
      <TricotNr>17</TricotNr>
      <Goals>0</Goals>
    </Player>
    <Player>
      <Name>Josip Brekalo</Name>
      <Birthday>23.06.1998</Birthday>
      <TricotNr>7</TricotNr>
      <Goals>0</Goals>
    </Player>
    <Player>
      <Name>Dawid Kownacki</Name>
      <Birthday>14.03.1997</Birthday>
      <TricotNr>9</TricotNr>
      <Goals>0</Goals>
    </Player>
    <Player>
      <Name>Luca Schuler</Name>
      <Birthday>22.03.1999</Birthday>
      <TricotNr>18</TricotNr>
      <Goals>0</Goals>
    </Player>
    <Player>
      <Name>Max Mustermann</Name>
      <Birthday>1.1.2001</Birthday>
      <TricotNr>99</TricotNr>
      <Goals>999</Goals>
    </Player>
    <Trainer>Stefan Leitl</Trainer>
    <Name>Hertha BSC</Name>
  </Team>
</Data>

```

## Ausgabe der Beispieldaten:

Liste der Mannschaften  
=====

Name : 1. FC Union Berlin  
Trainer : Steffen Baumgart  
Anzahl Spieler : 22  
Spieler:

01. Frederik Roennow (1; \* 04.08.1992)
02. Matheo Raab (31; \* 18.12.1998)
03. Danilho Doekhi (5; \* 30.06.1998)
04. Diogo Leite (4; \* 23.01.1999)
05. Leopold Querfeld (14; \* 20.12.2003)
06. Tom Rothe (15; \* 29.10.2004)
07. Josip Juranovic (18; \* 16.08.1995)
08. Christopher Trimmel (28; \* 24.02.1987)
09. Rani Khedira (8; \* 27.01.1994)
10. Aljoscha Kemlein (6; \* 02.08.2004)
11. Janik Haberer (19; \* 02.04.1994)
12. Andrej Ilic (23; \* 03.04.2000)
13. David Preu (45; \* 26.10.2004)
14. Woo-yeong Jeong (11; \* 20.09.1999)
15. Andras Schaefer (13; \* 13.04.1999)
16. Tim Starke (21; \* 07.09.1996)
17. Robert Skov (24; \* 20.05.1996)
18. Alex Kral (33; \* 19.05.1998)
19. Oliver Burke (7; \* 07.04.1997)
20. Livan Burcu (9; \* 28.09.2004)
21. Ilyas Ansah (10; \* 08.11.2004)
22. Dmytro Bogdanov (30; \* 06.03.2007)

Name : Hertha BSC  
Trainer : Stefan Leitl  
Anzahl Spieler : 23  
Spieler:

01. Tjark Ernst (1; \* 15.03.2003)
02. Marius Gersbeck (35; \* 20.06.1995)
03. Tim Goller (43; \* 26.01.2005)
04. Marton Dardai (31; \* 12.02.2002)
05. Linus Gechter (44; \* 27.02.2004)
06. John Brooks (25; \* 28.01.1993)
07. Toni Leistner (37; \* 19.08.1990)
08. Deyovaisio Zeefuik (42; \* 11.03.1998)
09. Michal Karbownik (33; \* 13.03.2001)
10. Julian Eitschberger (2; \* 05.03.2004)
11. Niklas Kolbe (27; \* 27.12.1996)
12. Tim Hoffmann (46; \* 26.02.2005)
13. Pascal Klemens (41; \* 23.02.2005)
14. Leon Jensen (5; \* 19.05.1997)
15. Paul Seguin (30; \* 29.03.1995)
16. Kennet Eichhorn (30; \* 27.07.2009)
17. Maurice Krattenmacher (14; \* 11.08.2005)
18. Kevin Sessa (8; \* 06.07.2000)
19. Boris Mamuzah Lum (21; \* 02.10.2007)
20. Michael Cuisance (10; \* 16.08.1999)
21. Diego Demme (6; \* 21.11.1991)
22. Dion Ajvazi (29; \* 10.02.2004)
23. Fabian Reese (11; \* 29.11.1997)