
Lehrveranstaltung "Algorithmen und Datenstrukturen" Übungsblatt 3

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 **22. Mai 2026**.

Aufgabe: In dieser Übungsaufgabe des Projektes "Mannschafts-Verwaltung" sollen zum Einen die Eingabe von neuen Mannschaften und neuen Spielern sowie zum Anderen die Ausgabe auf dem Bildschirm erstellt werden.

Für die Eingabe einer neuen Mannschaft (Funktion `createTeam` im Modul `teams.c`) soll der Benutzer die Namen von Mannschaft und Trainer eingeben; die Anzahl der Spieler wird erst einmal auf 0 gesetzt. In einer Schleife werden dann die Spieler erfasst (Funktion `createPlayer` im Modul `teams.c`; diese Funktion wird nur zur Eingabe **eines** Spielers verwendet!). Hierbei werden für einen Spieler jeweils Name, Geburtsdatum (Datumseingabe aus der 1. Übungsaufgabe) und die Trikotnr. erfasst; die Anzahl der bereits geschossenen Tore soll auf 0 gesetzt werden. Nach der erfolgreichen Eingabe eines Spielers (also wenn die Funktion `createPlayer` abgeschlossen ist) muss die Anzahl der Spieler in der Mannschaft um eins erhöht und der Benutzer gefragt werden, ob noch ein weiterer Spieler eingegeben werden soll (Funktion `askYesOrNo` im Modul `tools.c`). Die Eingabe von Trainernamen und Spielergeburtstag sind optional; alle anderen Eingaben sind Pflichteingaben.

Natürlich sollte geprüft werden, dass nicht mehr Mannschaften und Spieler eingegeben werden können als die Arrays dafür auch Platz haben.

Entsprechend der Beispielausgaben soll die Ausgabe aller Mannschaften (Funktion `listTeams` im Modul `teams.c`) erstellt werden. Für eine bestmögliche Wiederverwendbarkeit sollte diese Funktion noch unterteilt werden: Die Funktion `listTeams` ruft in einer Schleife für alle Mannschaften die Funktion `listOneTeam` auf, in der alle Daten einer Mannschaft auf dem Bildschirm ausgegeben werden. In dieser Funktion wird in einer Schleife über alle Spieler der Mannschaft die Funktion `listOnePlayer` aufgerufen. Hier werden in einer Zeile die Daten eines Spielers (Name, Trikotnr. und – sofern vorhanden – Geburtsdatum; siehe Beispielausgabe) ausgegeben.

Im Modul `datetime.c` werden für die Ein- und Ausgabe des Geburtsdatums folgende Funktionen hinzugefügt:

- `getDate`
- `printDate`

Die Funktion `getDate` kann teilweise aus der `ueb01.c` der 1. Übungsaufgabe übernommen werden. Beachten Sie aber, dass in der Datenstruktur nur ein Zeiger auf die Datumsstruktur gespeichert wird. D.h. es muss in dieser Funktion auch noch Speicher für die Datumsstruktur reserviert werden – und dies auch nur dann, wenn das Datum gültig ist.

Die Funktion `printDate` erhält einen Zeiger auf ein Datum vom Typ `sDate` und gibt dieses im Format `dd.mm.yyyy` auf dem Bildschirm aus (Tag und Monat zweistellig und das Jahr vierstellig jeweils mit führenden Nullen).

Da in den Datenstrukturen die Zeichenketten immer als Zeiger auf `char` und nicht Array von `char` definiert sind, müssen für alle Zeichenketten jeweils Speicherbereiche entsprechend der Eingabelängen reserviert werden. Dazu soll im Modul `tools.c` eine Funktion `getText` definiert werden, die alle hierfür notwendigen Aufgaben für die Eingabe eines Textes durchführt. Diese Funktion erhält eine Eingabeaufforderung als Zeichenkette, die maximale Eingabelänge für den einzugebenden Text, einen Wahrheitswert, ob auch ein leerer String als Eingabe zulässig ist sowie einen Zeiger auf eine Zeichenkette (also ein Zeiger auf Zeiger auf `char`); dieser Zeiger auf Zeichenkette soll jeweils auf die Zeichenketten in der Datenstruktur zeigen. In der Funktion soll ein Text mittels `scanf` in eine lokale, dynamisch reservierte Zeichenkette (Länge dieser Zeichenkette ist die als Parameter angegebene maximale Eingabelänge) eingelesen werden (diese muss am Ende der Funktion wieder freigegeben werden!). Dann soll entsprechend der Länge des eingegebenen Textes Speicher reserviert werden (über den Parameter Zeiger auf Zeichenkette sollen die Zeichenketten in den Datenstrukturen auf diesen jeweils reservierten Speicherbereich zeigen!). Dann wird der eingegebene Text in den reservierten Speicher kopiert. Die Funktion soll eine 1 zurückgeben, wenn alles geklappt hat, sonst eine 0. Damit soll mit dem Aufruf

```
getText("Name: ", 50, 0, &((Teams + TeamCounter)->Name));
```

einen Text eingelesen (max. 50 Zeichen; eine leere Eingabe ist nicht zulässig) und der Zeiger auf diesen Text in der Datenstruktur gespeichert werden können. Das Ermitteln der Textlänge sowie das Kopieren der Texte können mittels der Funktionen `strlen` und `strncpy` aus der `string.h` erfolgen.

Analog zur Funktion `getText` soll im Modul `tools.c` auch noch eine Funktion `getNumber` für die Eingabe von Zahlen definiert werden. Hierbei braucht aber kein Speicher reserviert werden. Mit zwei zusätzlichen Parametern `Von` und `Bis` kann ein Zahlenbereich angegeben werden, der von der Funktion auch gleich geprüft wird. Diese Funktion könnte z.B. wie folgt aufgerufen werden:

```
getNumber("Trikot-Nr.: ", &((Teams + TeamCounter)->TricotNr), 1, 99);
```

Die Funktion würde dabei erst dann verlassen werden, wenn der Benutzer eine gültige Zahl zwischen 1 und 99 eingegeben hat.

Generell soll immer mit Zeigern anstelle von Arrays gearbeitet werden! Oder anders ausgedrückt: Das Verwenden von eckigen Klammern ist tabu.

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).

Beispielausgabe:

Die Bildschirmausgabe soll folgendermaßen aussehen (die Benutzereingaben sind grau hinterlegt):

```
Mannschaften-Verwaltung V0.3
=====

1. Neue Mannschaft anlegen
2. Mannschaft loeschen
3. Suchen
4. Sortieren
5. Auflisten
6. Programm beenden

Ihre Wahl: 1
<Hier wird der Bildschirm gelöscht.>
Erfassung einer neuen Mannschaft
=====

Geben Sie bitte den Namen der Mannschaft ein:
-> 1. FC Union Berlin
Geben Sie bitte den Namen des Trainers ein:
-> Steffen Baumgart

Erfassung der Spieler
-----
Geben Sie bitte den Namen des Spielers ein:
-> Frederik Roennow
Geben Sie bitte das Geburtsdatum des Spielers ein:
->
Geben Sie bitte die Trikotnr. des Spielers ein:
-> 1

Moechten Sie einen weiteren Spieler eingeben (j/n) ? Jau

Geben Sie bitte den Namen des Spielers ein:
-> Leopold Querfeld
Geben Sie bitte das Geburtsdatum des Spielers ein:
-> 20.12.2003
Geben Sie bitte die Trikotnr. des Spielers ein:
-> 14

Moechten Sie einen weiteren Spieler eingeben (j/n) ? nö
<Hier wird der Bildschirm gelöscht.>
```

Mannschaften-Verwaltung V0.3

=====

1. Neue Mannschaft anlegen
2. Mannschaft loeschen
3. Suchen
4. Sortieren
5. Auflisten
6. Programm beenden

Ihre Wahl: 5

<Hier wird der Bildschirm gelöscht.>

Liste der Mannschaften

=====

Name : 1. FC Union Berlin
Trainer : Steffen Baumgart
Anzahl Spieler : 2
Spieler:
01. Frederik Roennow (1)
02. Leopold Querfeld (14; * 20.12.2003)

Name : Hertha BSC
Trainer : <dieser Posten ist vakant>
Anzahl Spieler : 2
Spieler:
01. Tjark Ernst (1; * 15.03.2003)
02. Diego Demme (6; * 21.11.1991)

Bitte Eingabetaste druecken ...