
Lehrveranstaltung "Algorithmen und Datenstrukturen" Übungsblatt 2

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 **01. November 2024 (Gruppe 2a/1a)** bzw. **08. November 2024 (Gruppe 2b/1b)**.

Aufgabe: In diesem Semester soll in den Übungsaufgaben eine Mannschafts-Verwaltung programmiert werden. Mit jeder Übungsaufgabe wird ein Teil des Programms erstellt, so dass am Ende des Semesters das komplette Programm fertig ist.

In der zweiten Übungsaufgabe soll das Projekt mit den Quellcode- und Headerdateien sowie die Datenstruktur erstellt werden. Auch sollen einige einfache Funktionen bereits erstellt werden.

Das Projekt soll jetzt aus folgenden Dateien bestehen – einige davon haben wir bereits in der ersten Übungsaufgabe erstellt:

<code>datastructure.h</code>	(Definition der Datenstrukturen)
<code>main.c</code>	(Hauptprogramm)
<code>teams.c + teams.h</code>	(Verwaltung der Mannschaften)
<code>datetime.c + datetime.h</code>	(Datumsfunktionen)
<code>menu.c + menu.h</code>	(Funktionalität für die Menüs)
<code>tools.c + tools.h</code>	(Hilfsfunktionen)

In der Headerdatei `datastructure.h` sollen die Datenstrukturen der Datenbank mit `typedef` definiert werden. Die Datenstrukturen sollen wie folgt aussehen (bitte alle Bezeichnungen auf Englisch).

- Datentyp `sDate`: Tag, Monat und Jahr als ganze Zahlen (sollte von der ersten Aufgabe bereits vorhanden sein)

- Datentyp `sPlayer`: Spielernamen als Zeiger auf Zeichen (kein `char-Array`!), Geburtsdatum vom Typ Zeiger auf `sDate` sowie die Trikotnr. und die Anzahl der geschossenen Tore als ganze Zahlen (keine Zeiger).

- Datentyp `sTeam`: Namen der Mannschaft und des Trainers als Zeiger auf Zeichen (auch hier keine `char-Arrays`!), die Anzahl der Spieler in der Mannschaft als ganze Zahl sowie die Spieler als Array vom Typ `sPlayer`. Das Array soll `MAXPLAYER` Elemente aufnehmen können;

`MAXPLAYER` ist eine Konstante mit dem Wert 23 und wird auch in der `datastructure.h` definiert.

Neben den Datenstrukturen soll hier mit der Speicherklasse `extern` auf die Variablen `TeamCounter` (ganze Zahl) und `Teams` (Array vom Datentyp `sTeam`) verwiesen werden; beide Variablen sollen im Modul `teams.c` als globale Variablen definiert werden.

Im Modul `team.c` werden folgende Funktionen definiert, die vom Hauptprogramm entsprechend der Menüauswahl aufgerufen werden sollen. Alle Funktionen haben keine Parameter und geben nichts zurück. Jede Funktion soll nur ihren Funktionsnamen auf dem Bildschirm ausgeben und anschließend auf das Drücken der Eingabetaste warten (siehe Funktion `waitForEnter` im Modul `tools.c`). Die eigentlichen Funktionalitäten kommen erst in den nächsten Übungsaufgaben.

- `createTeam`
- `deleteTeam`
- `searchPlayer`
- `sortTeams`
- `listTeams`

Ferner sollen hier die globalen Variablen (siehe oben) `TeamCounter` und ein Array namens `Teams` vom Typ `sTeam` mit `MAXTEAMS` Elementen definiert werden; `MAXTEAMS` ist eine Konstante mit dem Wert 10 und wird in der Headerdatei `datastructure.h` definiert.

Im Modul `menu.c` wird folgende Funktion definiert:

- `getMenu`

Diese Funktion erhält eine Zeichenkette mit dem Menütitel und ein Array von Zeichenketten, in denen die Menüpunkte stehen (so wie sie auf dem Bildschirm erscheinen sollen) sowie die Anzahl der Menüpunkte. D.h. die Texte des Titels und der Menüpunkte werden vom Aufrufer (hier die `main`-Funktion) vorgegeben; dadurch können wir diese Funktion später auch für Untermenüs verwenden. Die Funktion soll den gewählten Menüpunkt als Zahl zurückgeben. In der Funktion wird in einer Schleife erst der Bildschirm gelöscht (siehe Funktion `clearScreen` im Modul `tools.c`), dann der Menütitel sowie die einzelnen Menüpunkte ausgegeben. Nach den Menüpunkten soll mit einer Leerzeile Abstand eine Eingabeaufforderung auf dem Bildschirm erscheinen. Dann darf der Benutzer die Nummer des gewünschten Menüpunktes eingeben. Die Schleife darf nur verlassen werden, wenn der Benutzer eine gültige Eingabe (zwischen 1 und der Anzahl der Menüpunkte) eingegeben hat.

Das Modul `tools.c` aus der ersten Übungsaufgabe soll um eine Funktion erweitert werden:

- `printLine`

Diese Funktion erhält ein Zeichen und eine Anzahl als Parameter und gibt nichts zurück. In der Funktion soll das angegebene Zeichen entsprechend der Anzahl mehrmals hintereinander ausgegeben werden. Z.B. wird mit dem Aufruf `printLine('-', 50)`; eine Linie von 50 Minuszeichen ausgegeben. Diese Funktion kann bei der Ausgabe der Menüs sowie der Auflistung der Datensätze gut verwendet werden.

In der Datei `main.c` wird nur die `main`-Funktion definiert. Mittels Include-Anweisungen werden die notwendigen Headerdateien mit den Funktionsprototypen der einzelnen Module eingefügt. In der `main`-Funktion werden erst die Texte für das Menü definiert (Array von Zeichenketten), dann wird in einer Schleife das Hauptmenü (Funktion `getMenu` im Modul `menu.c`) aufgerufen und dem Ergebnis entsprechend die dazugehörige Funktion aus dem Modul `teams.c` aufgerufen. Die Schleife darf erst dann verlassen werden, wenn der Benutzer den Menüpunkt Programmende gewählt hat.

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 V0.2
```

```
=====
```

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

```
Ihre Wahl: 1  
createTeam
```

```
Bitte Eingabetaste druecken ...
```