
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! Compilerwarnungen sollen möglichst vermieden werden.).

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 **02. November 2023**.

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

In der zweiten Übungsaufgabe sollen die 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:

datastructure.h (hier werden die Datenstrukturen definiert)
main.c (Hauptprogramm)
calendar.c (Funktionen für die Verwaltung der Termine)
calendar.h (dazugehörige Headerdatei)
menu.c (Funktionalität für die Menüs)
menu.h (dazugehörige Headerdatei)
datetime.c (Datums- und Uhrzeitfunktionen)
datetime.h (dazugehörige Headerdatei)
tools.c (Hilfsfunktionen)
tools.h (dazugehörige Headerdatei)

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

- Datentyp `eDayOfTheWeek`: Aufzählungstyp mit den Werten `NotADay`, `Mo`, `Tu`, `We`, `Th`, `Fr`, `Sa` und `Su`, wobei `NotADay` den Wert 0, `Mo` den Wert 1, `Tu` den Wert 2 usw. hat.

- Datentyp `sDate`: Tag, Monat und Jahr als ganze Zahlen sowie den Wochentag vom Typ `eDayOfTheWeek`. Hier muss also der Datentyp von der ersten Übungsaufgabe um den Wochentag erweitert werden.
- Datentyp `sTime`: Stunden, Minuten und Sekunden als ganze Zahlen.
- Datentyp `sAppointment`: Datum und Uhrzeit; siehe die oben definierten Datentypen `sDate` und `sTime`, je eine Zeichenkette für die Beschreibung und für den Ort (Zeiger auf `char`; kein `char-Array`!) und eine Termindauer als Zeiger auf `sTime`.

Mittels `#define` bzw. `enum` wird die maximale Anzahl von Termineinträgen (`MAXAPPOINTMENTS`) als Konstante festgelegt, z.B. auf 100.

Mit der Speicherklasse `extern` wird auf die Variablen `countAppointments` (ganze Zahl) und `Calendar` (Array vom Datentyp `sAppointment`) verwiesen. Damit werden diese Variablen nur deklariert, aber nicht definiert!

In der Datei `calendar.c` werden folgende Funktionen definiert, die vom Hauptprogramm entsprechend aufgerufen werden sollen. Im ersten Schritt haben alle Funktionen keine Parameter und geben `void` zurück. Jede Funktion soll nur den Namen der Funktion oder des Menüs auf dem Bildschirm ausgeben und dann auf die Eingabetaste warten (Funktion `waitForEnter`). Die eigentliche Funktionalität kommt erst in den nächsten Übungsaufgaben.

- `createAppointment`
- `editAppointment`
- `deleteAppointment`
- `searchAppointment`
- `sortCalendar`
- `listCalendar`

Ferner sollen hier die globalen Variablen `countAppointments` (ganze Zahl) und `Calendar` (Array vom Datentyp `sAppointment` mit `MAXAPPOINTMENTS` Elementen) definiert werden (zur Erinnerung: diese wurden in der `datastructure.h` deklariert).

In der Datei `menu.c` wird folgende Funktion definiert:

- `getMenu`

Diese Funktion erhält eine Zeichenkette mit dem Menütitel, der über dem Menü stehen soll, ein Array von Zeichenketten, in denen die Menüpunkte stehen (so wie sie auf dem Bildschirm erscheinen sollen) sowie die Anzahl der Menüpunkte. Die Funktion soll den gewählten Menüpunkt als Zahl zurückgeben. In der Funktion wird in einer Schleife erst der Bildschirm gelöscht, dann der Titel sowie die einzelnen Menüpunkte ausgegeben. Nach den Menüpunkten soll mit einer Leerzeile Abstand eine Eingabeaufforderung auf dem Bildschirm erscheinen. Die Schleife darf nur verlassen werden, wenn der Benutzer eine gültige Eingabe (zwischen 1 und der Anzahl der Menüpunkte) eingegeben hat.

Wichtig: Diese Funktion soll allgemein gelten, d.h. diese Funktion soll später auch für Untermenüs verwendet werden können.

In der Datei `tools.c` werden folgende Funktionen definiert:

```
- clearBuffer
- waitForEnter
- clearScreen
- askYesOrNo
- printLine
```

Zu den bereits bekannten Funktionen kommt nun noch die Funktion `printLine`. Sie erhält ein Zeichen und eine Anzahl und gibt `void` zurück. In der Funktion soll das angegebene Zeichen entsprechend der Anzahl mehrmals hintereinander ausgegeben werden. Mit dem Aufruf `printLine('-', 50);` wird z.B. eine Linie von 50 Minuszeichen ausgegeben. Diese Funktion kann bei der Ausgabe der Menüs sowie der Datensätze gut verwendet werden.

In der Datei `main.c` wird nur die `main`-Funktion definiert. Mittels Include-Anweisungen werden die Funktionsprototypen (Headerdateien) eingefügt. In der `main`-Funktion wird in einer Schleife das Hauptmenü aufgerufen (Funktion `getMenu`; die Menüpunkte werden hier lokal im Hauptprogramm in Form eines Arrays von Zeichenketten definiert und dann an die Funktion übergeben); anschließend wird entsprechend des gewählten Menüpunktes (`switch`-Anweisung) die dazugehörige Funktion aus dem Modul `calendar.c` aufgerufen. Dies wird solange wiederholt, bis der Menüpunkt „Programm beenden“ aufgerufen wird.

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).

Die Bildschirmausgabe soll folgendermaßen aussehen:

```
Terminverwaltung V 0.2
=====

1. Neuen Termin anlegen
2. Termin bearbeiten
3. Termin loeschen
4. Termin suchen
5. Termine sortieren
6. Termine auflisten
7. Programm beenden

Ihre Wahl: 1
ToDo: Termin erzeugen

Bitte Eingabetaste druecken ...
```