



Lehrveranstaltung "Informatik II für TI-Bachelor"

Übungsblatt 2

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 **26. April 2013**.

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

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 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.

Folgende Dateien sollen erstellt werden:

- `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)
- `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 `TTime`: Stunden und Minuten als ganze Zahlen.

Datentyp `TDate`: Tag, Monat und Jahr sowie den Wochentag als ganze Zahlen (Sonntag = 0, Montag = 1, usw.).

Datentyp `TAppointment`: Uhrzeit und Datum als Zeiger auf die oben definierten Datentypen `TTime` und `TDate`, eine Zeichenkette für die

Beschreibung (Zeiger auf `char`; kein `char`-Array!) und eine Termindauer als Zeiger auf `TTime`.

Mittels `#define` wird die maximale Anzahl von Termineinträgen (`MAX-APPOINTMENTS`) festgelegt, z.B. auf 100.

Mit der Speicherklasse `extern` wird auf die Variablen `AppointmentCount` (ganze Zahl) und `Calendar` (Array vom Datentyp `TAppointment`) verwiesen. Damit werden diese Variablen 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 `AppointmentCount` (ganze Zahl) und `Calendar` (Array vom Datentyp `TAppointment` mit `MAX-APPOINTMENTS` Elementen) definiert werden (zur Erinnerung: diese wurden in der `datastructure.h` deklariert).

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

- `Menu`

Diese Funktion erhält eine Zeichenkette mit dem Titel, 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.

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

- `clearBuffer`
- `waitForEnter`
- `clearScreen`
- `askAgain`
- `printLine`

Zu den bereits bekannten Funktionen kommt nun noch die Funktion `printLine`. Sie erhält ein Zeichen und eine Anzahl und gibt ein `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 `Menu`; die Menüpunkte werden hier 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.

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 1.0
=====

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
createAppointment

Bitte Eingabetaste druecken ...
```