



Lehrveranstaltung "Informatik II für TI-Bachelor"

Übungsblatt 3

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 **17. Mai 2013**.

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

Aufgabe: In der dritten Übungsaufgabe sollen für das Projekt „Termin-Verwaltung“ die Eingabe von neuen Terminen sowie die Ausgabe der Termine auf dem Bildschirm erstellt werden.

Die Datenstruktur für einen Termin `TAppointment` soll noch um ein Feld erweitert werden: `Location` vom Typ `char *`.

Für die Eingabe eines neuen Termins (Funktion `createAppointment` im Modul `calendar.c`) soll der Benutzer Datum, Uhrzeit, Terminbeschreibung, Ort und die Dauer eingeben können.

Für die Eingabe eines Datums passen Sie die Funktionen der ersten Übungsaufgabe an: Die Funktion `getDate` erhält neben der Eingabeaufforderung jetzt nicht mehr Zeiger auf Tag, Monat und Jahr, sondern einen Zeiger auf den Datumszeiger aus der Datenstruktur eines Termins. Die Funktion soll bei erfolgreicher Datumseingabe Speicher für die Datumsstruktur reservieren und dort das eingegebene Datum eintragen. Ferner soll der Wochentag ermittelt werden (siehe nächsten Absatz). Entsprechend werden die anderen Funktionen der ersten Übungsaufgabe angepasst und erweitert.

Um den Wochentag für ein Datum zu ermitteln, wird die Summe aus fünf Zahlen berechnet:

Zahl 1: Tagesziffer -> Divisionsrest vom Tag im Monat durch 7.

Zahl 2: Monatsziffer -> Die Anzahl der Tage, um die sich der Wochentag bei den verschiedenen Monaten verschiebt:

Jan.: 0	Feb.: 3	Mrz.: 3	Apr.: 6	Mai: 1	Jun.: 4
Jul.: 6	Aug.: 2	Sep.: 5	Okt.: 0	Nov.: 3	Dez.: 5

Zahl 3: Jahresziffer -> Jahr im Jahrhundert + (Jahr im Jahrhundert dividiert durch 4 und davon der Divisionsrest durch 7.

Zahl 4: Jahrhundertziffer -> $(3 - (\text{Jahrhundert modulo } 4)) \text{ mal } 2$ (Dabei besteht das Jahrhundert aus den ersten beiden Ziffern des Jahres).

Zahl 5: Schaltjahreskorrektur -> Liegt das Datum im Januar oder Februar und ist das Jahr ein Schaltjahr, muss für die Summe noch 1 abgezogen werden.

Der Divisionsrest von der Summe dieser fünf Zahlen durch 7 ergibt eine Zahl von 0 bis 6. Dabei ist die 0 Sonntag, 1 Montag, usw.

Weitere Informationen zur Wochentagsberechnung finden Sie z.B. bei Wikipedia unter dem Stichwort „Wochentagsberechnung“.

Für die Eingabe einer Uhrzeit schreiben Sie ebenfalls eine eigene Funktion (`getTime`) im Modul `dateutils.c` (analog zu `getDate`): Eingabe in lokale Zeichenkette, Parsen der Uhrzeit in Stunden und Minuten, Prüfen auf gültige Uhrzeit, Speicher reservieren und die eingegebene Uhrzeit dort eintragen.

Diese Funktion wird dann auch für die Eingabe der Termindauer verwendet.

Für die Eingaben Terminbeschreibung und Ort wird eine Funktion `getText` benötigt, die genauso viel Speicher reserviert, wie der eingegebene Text lang ist (diese Funktion wurde in der Vorlesung am 19.04. vorgestellt). Als Parameter erhält sie eine Eingabeaufforderung (Zeichenkette), einen Zeiger auf die Zeichenkette in der Datenstruktur (Terminbeschreibung bzw. Ort) sowie die maximale Länge der Texteingabe. Dabei soll die Terminbeschreibung max. 100 Zeichen und der Ort max. 15 Zeichen lang sein. In der Funktion wird zuerst Speicher für eine Zeichenkette mit der maximalen Textlänge reserviert. Nachdem der Benutzer dort den Text eingegeben hat, wird für den Zeiger auf die Zeichenkette (2. Parameter) entsprechend der eingegebenen Textlänge (`strlen`) Speicher reserviert und dort hinein der Text kopiert (`strcpy`). Der zuerst reservierte Speicherbereich über die maximale Textlänge muss am Ende wieder freigegeben werden.

Die Listenausgabe aller Termine wird in der Funktion `listCalendar` programmiert. Nach der Ausgabe einer Listenüberschrift werden in einer Schleife alle Termine tabellarisch ausgegeben (siehe auch Beispielausgabe). Dabei wird zuerst das Datum ausgegeben und darunter alle Termine dieses Datums (Gruppierung nach Datum; richtig sinnvoll wird dies erst später, wenn die Termine chronologisch sortiert sind!). Die Ausgabe eines Termins erfolgt in einer eigenen Funktion (damit später auch mal ein einzelner Termin ausgegeben werden kann); der Termin wird als Zeiger übergeben. Für die formatierte Ausgabe von Datum und Uhrzeit werden entsprechende Funktionen `printDate` und `printTime` im Modul `dateutils.c` erstellt (Ausgabeformat `dd.mm.yyyy` und `hh:mm`, jeweils mit führenden Nullen). Von der Terminbeschreibung sollen maximal 50 Zeichen ausgegeben werden; ist der Text länger als 50 Zeichen, werden nach den 50 Zeichen noch drei Punkte ausgegeben.

Nach der Ausgabe von z.B. jeweils 15 Terminen soll auf das Drücken der Eingabetaste gewartet werden (Funktion `waitForEnter`).

Auf keinen Fall dürfen Sie vergessen, bei Programmende alle reservierten Speicherbereiche wieder freizugeben! Dazu erstellen Sie

eine Funktion `freeCalendar`, in der in einer Schleife jeweils der Speicher eines Termins freigegeben wird (Funktion `freeAppointment`).

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 für Termineingabe und -ausgabe soll folgendermaßen aussehen:

Erfassung eines neuen Termins

=====

```
Datum       : 26.4.2013
Uhrzeit     : 8:00
Beschreibung : Informatik II - Gruppe 1 - Abgabe der zweiten Uebungsaufgabe
Ort         : Raum D113
Dauer       : 1:30
Termin wurde gespeichert!
```

Bitte Eingabetaste druecken ...

Liste der Termine

=====

Freitag, der 26.04.2013:

```
08:00 -> Raum D113      | Informatik II - Gruppe 1 - Abgabe der zweiten Uebu ...
10:00 -> Raum H5        | Informatik II - Vorlesung
12:15 -> Raum D113      | Informatik II - Nachsitzen
```

Mittwoch, der 01.05.2013:

```
20:00 -> zuhause        | Feier zum 1. Mai
```

Freitag, der 03.05.2013:

```
10:00 -> Raum H5        | Informatik II - Vorlesung
```

Bitte Eingabetaste druecken ...