
Lehrveranstaltung "Algorithmen und Datenstrukturen" Übungsblatt 4

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 **30. November 2023**.

Aufgabe: In der vierten Übungsaufgabe sollen in der Terminverwaltung die Daten beim Beenden des Programms in einer Datei gespeichert werden, damit sie beim nächsten Programmstart wieder eingelesen werden können.

Für das Laden und Speichern der Daten sollen einige Funktionen in einem neuen Modul `database.c` erstellt werden. Erzeugen Sie das Modul samt Headerdatei und passen Sie das Projekt an.

Für das Speichern der Daten soll im neuen Modul die Funktion `saveCalendar` geschrieben werden. Hier soll nach dem erfolgreichen Öffnen der Datei die erste Zeile mit der Startkennung der Daten (`<Calendar>`; siehe Beispiel) geschrieben werden. Dann werden in einer Schleife die Daten der einzelnen Termine geschrieben; das Schreiben eines Termins kann eine Funktion namens `saveAppointment` übernehmen. Nach der Startkennung eines Termins (`<Appointment>`) werden die Daten des Termins (Datum, Uhrzeit, Beschreibung, Ort und Dauer) jeweils mit Start- und Endkennung in jeweils eine Zeile geschrieben (z.B: `<Date>19.11.2021</Date>`). Nach den Feldern eines Termins wird die Endkennung (`</Appointment>`) in eine eigene Zeile geschrieben. Nachdem alle Termine auf dieser Art geschrieben wurden, wird noch die Endkennung der Daten (`</Calendar>`) geschrieben.

Für das Laden der Daten soll eine Funktion `loadCalendar` geschrieben werden. In dieser Funktion sollen nach dem erfolgreichen Öffnen der Datei die Daten eingelesen werden. Immer wenn die Startkennung eines Termins gefunden wird, soll die Funktion `loadAppointment` aufgerufen werden. Diese liest die Daten dieses einen Termins ein (bis zur Endkennung `</Appointment>`).

Durch die Start- und Endkennungen können die Felder in beliebiger Reihenfolge stehen. Ferner kann darüber beim Einlesen geprüft werden, ob die Struktur der Daten gültig ist. Um die Datenbank kontrollieren zu können, sollte auf eine Verschlüsselung der Daten verzichtet werden. Überlegen Sie sich, welche Maßnahmen noch nötig sind, um angemessen auf eine fehlerhafte Datenbank zu reagieren. Und was muss getan werden, wenn in einem Termin-Datensatz ein Feld fehlt (z.B: in der Beispieldatenbank fehlen bei mehreren Terminen verschiedene Felder), damit dieser Datensatz trotzdem noch korrekt eingelesen und angezeigt werden kann.

Wem diese ganze Beschreibung zu kompliziert ist, guckt sich am besten erst das unten stehende Beispiel an; dies sollte vieles erklären.

Noch ein paar Hinweise:

1. Nach dem Laden und Speichern sollte die Datei natürlich jeweils wieder geschlossen werden.
2. Das Laden der Daten erfolgt bei Programmstart; das Speichern (optional mit Benutzerabfrage) bei Programmende. Wer möchte, kann alternativ Laden und Speichern als Menüpunkte einbauen (das Laden sollte dann die Daten zu den bereits eingegebenen Daten hinzufügen!).
3. Vor dem Einlesen eines jeden Datensatzes sollten die Felder mit Nullen (0 bzw. NULL) gefüllt werden.
4. Bei Programmende müssen natürlich (sofern nicht bereits in der vorigen Übungsaufgabe erledigt) alle reservierten Speicherbereiche wieder freigegeben werden!
5. Beim Einlesen kann immer eine ganze Zeile eingelesen werden. Um herauszufinden, welche Daten in der Zeile stehen (dieser Vorgang wird „parsen“ genannt), kann z.B. die Funktion `strncmp` aus der `string.h` verwendet werden; z.B. wird mit

```
if (strncmp(Zeile, "<Date>", 6) == 0)
```

geprüft, ob die ersten 6 Zeichen in der Zeichenkette `Zeile` gleich `<Date>` sind.
6. An Anfang einer Zeile können beliebig viele Leerzeichen oder Tabulatoren stehen.
7. Innerhalb eines Datensatzes können Tags mehrfach vorkommen; entsprechend müssen die Speicherbereiche der bereits eingelesenen Daten wieder freigegeben werden.

Generell soll immer möglichst mit Zeigern anstelle von Arrays gearbeitet werden!

Kommentieren Sie das Programm. Dazu gehören auch Modul- und Funktionsheader (siehe Skript „Grundlagen der Informatik“ Kapitel 5.3 und 5.4)! Achten Sie auch auf Ihre Programmstruktur (Einrückungen; Leerzeichen und Leerzeilen, usw.).

Beispieldatenbank

```
<Calendar>
  <Appointment>
    <Date>30.11.2023</Date>
    <Time>15:10:00</Time>
    <Description>Ausschlafen</Description>
    <Description>zur Hochschule fahren</Description>
  </Appointment>
  <Appointment>
    <Date>30.11.2023</Date>
    <Time>16:00:00</Time>
    <Description>AuD Seminaristischer Unterricht</Description>
    <Duration>01:30</Duration>
    <Location>Raum D 136 (H5)</Location>
  </Appointment>
  <Appointment>
    <Duration>01:45</Duration>
    <Location>Raum D 114</Location>
    <Description>AuD Uebungsgruppe 2 - Vierte Uebungsaufgabe abgeben</Description>
    <Time>17:45:00</Time>
    <Date>30.11.2023</Date>
  </Appointment>
  <Appointment>
    <Location>Raum D 114</Location>
    <Date>30.11.2023</Date>
    <Time>19:30:0</Time>
    <Description>AuD Nachsitzen</Description>
    <Duration>1:0</Duration>
  </Appointment>
  <Appointment>
    <Date>6.12.2023</Date>
    <Time>4:0</Time>
    <Description>dem Nikolaus auflauern</Description>
    <Location>Wohnzimmer</Location>
    <Duration>2:00</Duration>
  </Appointment>
  <Appointment>
    <Date>31.12.2023</Date>
    <Time>20:00</Time>
    <Description>Silvester-Party</Description>
    <Location></Location>
    <Duration>08:00</Duration>
  </Appointment>
  <Appointment>
    <Date>4.1.2024</Date>
    <Time>16:00</Time>
    <Description>AuD - endlich wieder Seminaristischer Unterricht</Description>
    <Location>Raum D 136 (H5)</Location>
    <Duration>01:30</Duration>
  </Appointment>
  <Appointment>
    <Date>4.1.2024</Date>
    <Time>17:45</Time>
    <Description>AuD - Abgabe 6. Aufgabe</Description>
    <Location>Raum D 114</Location>
    <Duration>01:30</Duration>
  </Appointment>
</Calendar>
```

Ausgabe der Beispieldatenbank:

Liste der Termine

=====

Do, 30.11.2023:

15:10 ->	zur Hochschule fahren
16:00 -> Raum D 136 (H5)	AuD Seminaristischer Unterricht
17:45 -> Raum D 114	AuD Uebungsgruppe 2 - Vierte Uebungsaufgabe ...
19:30 -> Raum D 114	AuD Nachsitzen

=====

Mi, 06.12.2023:

04:00 -> Wohnzimmer	dem Nikolaus auflauern
---------------------	------------------------

=====

So, 31.12.2023:

20:00 ->	Silvester-Party
----------	-----------------

=====

Do, 04.01.2024:

16:00 -> Raum D 136 (H5)	AuD - endlich wieder Seminaristischer Unterricht
17:45 -> Raum D 114	AuD - Abgabe 6. Aufgabe