
Lehrveranstaltung "Algorithmen und Datenstrukturen" Übungsblatt 3

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 **16. November 2023**.

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.

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 und einer Uhrzeit erstellen Sie im Modul `datetime.c` die Funktionen `getDate` und `getTime`. Diese sollen als Parameter eine Eingabeaufforderung und einen Zeiger auf die Datums- bzw. Zeitstruktur erhalten. Der Benutzer kann nach der Ausgabe der Eingabeaufforderung ein Datum bzw. eine Uhrzeit eingeben. Die Eingabe wird dann mittels der Funktionen `getDateFromString` bzw. `getTimeFromString` in die Datenstrukturen umgewandelt. Die Funktionalität können Sie aus dem Hauptprogramm der ersten Übungsaufgabe übernehmen und entsprechend anpassen.

Die `getTime`-Funktion soll auch für die Eingabe der Termindauer verwendet werden. Hier muss aber zusätzlich nach der erfolgreichen Zeiteingabe Speicher für die Termindauer reserviert werden und dorthin die Zeit übertragen werden, da in der Datenstruktur nur ein Zeiger für die Dauer vorhanden ist. Wird keine Zeitdauer angegeben (also eine leere Eingabe), dann muss als Dauer ein `NULL`-Zeiger gespeichert werden.

Die Funktion `getDateFromString` muss noch um die Ermittlung des Wochentages erweitert werden. Die dazu benötigte Formel finden Sie z.B. bei Wikipedia unter dem Stichwort „Wochentagsberechnung“.

Für die Eingaben Terminbeschreibung und Ort wird eine Funktion `getText` (im Modul `tools.c`) benötigt, die genau soviel Speicher reserviert, wie der eingegebene Text lang ist. Als Parameter erhält sie eine Eingabeaufforderung (Zeichenkette), die maximale Länge der Texteingabe, einen

Zeiger auf die Zeichenkette in der Datenstruktur (Terminbeschreibung bzw. Ort) sowie einen Wahrheitswert, ob auch eine leere Eingabe erlaubt ist (die Terminbeschreibung ist eine Pflichteingabe; die Eingabe des Ortes ist optional). 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 (3. 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; zur Zeit müssen die Termine vom Benutzer in chronologischer Reihenfolge eingegeben werden!). 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 `datetime.c` erstellt (Ausgabeformat `dd.mm.yyyy` und `hh:mm`, jeweils mit führenden Nullen). Von der Terminbeschreibung sollen maximal 48 Zeichen ausgegeben werden; ist der Text länger als 48 Zeichen, werden nur die ersten 44 Zeichen und dahinter noch ein Leerzeichen und 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 (z.B. 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 (Benutzereingaben sind grau hinterlegt):

Erfassung eines neuen Termins

=====

Datum : 16.11.2023
Uhrzeit : 17:45
Beschreibung : AuD - Übungsgruppe 1 - Abgabe der zweiten Übungsaufgabe
Ort : Raum D114
Dauer : 0:45
Termin wurde gespeichert!

Bitte Eingabetaste druecken ...

Liste der Termine

=====

Do, 16.11.2023:

| | |
|--------------------|--|
| 15:10 -> | zur Hochschule fahren |
| 16:00 -> Raum D136 | AuD - Seminaristischer Unterricht |
| 17:45 -> Raum D114 | AuD - Übungsgruppe 1 - Abgabe der zweiten Üb ... |
| 19:30 -> Raum D114 | AuD - Nachsitzen |

So, 31.12.2023:

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

Do, 04.01.2024:

| | |
|--------------------|--|
| 16:00 -> Raum D136 | AuD - Seminaristischer Unterricht |
| 17:45 -> Raum D114 | AuD - Übungsgruppe 1 - Abgabe der sechsten Ü ... |

Bitte Eingabetaste druecken ...