Dipl.Phys. Gerald Kempfer

Berliner Hochschule für Technik
- University of Applied Sciences -
Fachbereich VI – Informatik und Medien
Studiengang Technische Informatik Bachelor

Lehrveranstaltung "Algorithmen und Datenstrukturen"
Übungsblatt 1

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!).

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 **24. April 2026**.

Aufgabe: Ziel der ersten Übung ist das Kennenlernen der Arbeitsmittel, die in diesem Semester benötigt werden: eine IDE – Integrierte Entwicklerumgebung wie z.B. Code::Blocks oder Visual Studio Code.

Schauen Sie sich das Video „Einführung in Code::Blocks“ an. Installieren Sie sich eine IDE Ihrer Wahl (Empfehlung: Code::Blocks bzw. XCode für Mac) und probieren Sie es auf Ihrem Rechner aus.

Organisieren Sie sich in Dreiergruppen (Zweier- oder Einergruppen sind auch möglich) – die Gruppen werden bei der Abgabe dieser Aufgabe von mir erfasst. Gemeinsam in dieser Gruppe bearbeiten Sie dann die Übungsaufgaben.

Erstellen Sie als erstes die Headerdatei `datastructure.h`. In dieser Datei soll eine Datenstruktur mittels `typedef` deklariert werden (möglichst alles in Englisch):
- `sDate`: beinhaltet drei Zahlen für Tag, Monat und Jahr

In dieser Headerdatei kommen später noch weitere Datenstrukturen hinzu.

Schreiben Sie dann die fehlenden Funktionen zum vorgegebenen Hauptprogramm. Diese Funktionen sollen in einem eigenen Modul (z.B. `datetime.c`) untergebracht werden, da diese in den weiteren Übungsaufgaben wieder benötigt werden.

Die Funktion `isLeapYear` (soll von der nächsten Funktion aufgerufen werden) soll als Funktionsergebnis eine ganze Zahl zurückgeben. Diese Zahl soll als Wahrheitswert angeben, ob das angegebene Jahr (Parameter) ein Schaltjahr ist.

Die Funktion `isDateValid` (soll von der nächsten Funktion aufgerufen werden) soll als Funktionsergebnis eine ganze Zahl zurückgeben. Diese Zahl soll als Wahrheitswert angeben, ob das angegebene Datum (Parameter Struktur `sDate` mit Tag, Monat und Jahr) ein gültiges Datum ist. Dabei sollen auch die Schaltjahre berücksichtigt werden.

Die Funktion `getDateFromString` (wird vom Hauptprogramm aufgerufen) soll als Funktionsergebnis eine ganze Zahl zurückgeben. Diese Zahl soll als Wahrheitswert angeben, ob in der angegebenen Zeichenkette (1. Parameter) ein gültiges Datum enthalten ist. Dieses Datum soll in Tag, Monat und Jahr (als Zahlen) zerlegt – sozusagen geparst – werden; dabei sind die Zahlen durch Punkte voneinander getrennt; die Punkte müssen nicht unbedingt an den Positionen 3 und 6 stehen (siehe Beispielausgabe)! Natürlich sollen diese drei Zahlen als Datum geprüft werden, ob sie ein gültiges Datum ergeben. Wenn ja, sollen diese drei Zahlen in die übergebene Datumsstruktur (2. Parameter: Zeiger auf `sDate`) gespeichert werden.

Ferner soll wieder wie im vorigen Semester ein Modul namens `tools.c` erstellt werden mit Hilfsfunktionen wie
- `clearBuffer()`
- `waitForEnter()`
- `clearScreen()` (hier wird die Funktion `system` aus der Headerdatei `stdlib.h` verwendet, z.B. `system("CLS");`) unter Windows bzw. `system("clear");` unter Linux und Mac).
- `askYesOrNo(char *Question)`

Dieses Modul wird im Laufe dieses Semesters immer wieder benötigt und wird noch erweitert werden.
Unter Linux kann auch wieder die Headerdatei `escapesequenzen.h` verwendet werden (dies funktioniert leider nicht unter Windows/Mac).

Es dürfen keine Headerdateien außer `stdio.h` und `stdlib.h` für diese erste Übung verwendet werden. Ferner darf die Funktion sscanf nicht verwendet werden (scanf darf verwendet werden).

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).

Bei der Abgabe soll die Funktion `getDateFromString` von Ihnen schrittweise im Debugger vorgeführt und dabei alle Variableninhalte angezeigt werden.

Beispielausgabe:

Die Bildschirmausgabe soll folgendermaßen aussehen
(die Benutzereingaben sind grau hinterlegt):

```
Geben Sie bitte ein gueltiges Datum ein: 29.2.2026
Das eingegebene Datum '29.2.2026' ist ungueltig!

Moechten Sie noch einmal (j/n) ? jau
<Hier wird der Bildschirm gelöscht!>
Geben Sie bitte ein gueltiges Datum ein: 29.2.2024
Das Datum 29.02.2024 ist gueltig!


Moechten Sie noch einmal (j/n) ? Ja
<Hier wird der Bildschirm gelöscht!>
Geben Sie bitte ein gueltiges Datum ein: 008.010.002026
Das Datum 08.10.2026 ist gueltig!


Moechten Sie noch einmal (j/n) ? jupp
<Hier wird der Bildschirm gelöscht!>
Geben Sie bitte ein gueltiges Datum ein: <Eingabetaste>
Das eingegebene Datum '' ist ungueltig!

Moechten Sie noch einmal (j/n) ? nö
```

Hauptprogramm:

```
#include <stdio.h>
#include "datastructure.h"
#include "datetime.h"
#include "tools.h"

void inputDate();

int main()
{
   do
   {
      clearScreen();
      inputDate();
   } while (askYesOrNo("Moechten Sie noch einmal (j/n) ? "));

   return 0;
}

/*********************************************************************
 * Funktion: void inputDate(void)
 *   - Benutzer soll ein Datum eingeben.
 *   - Eingabe wird mit Hilfe der Funktion getDateFromString geparst
 *     und geprueft. Bei gueltigem Datum steht dieses in der Datums-
 *     variable Date.
 *   - Ergebnis der Eingabe wird entsprechend angezeigt.
 * Paramater: keine
 * Funktionsergebnis: nichts
 *********************************************************************/
void inputDate()
{
   sDate Date;
   char Input[20];

   printf("Geben Sie bitte ein gueltiges Datum ein: ");
   *Input = '\0';
   scanf("%19[^\n]", Input);
   clearBuffer();
   if (getDateFromString(Input, &Date))
      printf("Das Datum %02i.%02i.%04i ist gueltig!\n", Date.Day, Date.Month, Date.Year);
   else
      printf("Das eingegebene Datum '%s' ist ungueltig!\n", Input);

   printf("\n");
}
```