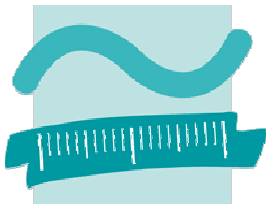
Dipl.Phys. Gerald Kempfer

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



Lehrveranstaltung "Informatik II für TI-Bachelor"
Übungsblatt 1

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 **12. April 2013**.

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

Aufgabe:

Ziel der ersten Übung ist das Kennenlernen der Arbeitsmittel, die in diesem Semester benötigt werden: PCs des LIS-Labors und die Entwicklerumgebung Code::Blocks.

Im ersten Teil werden wir den Zugang zu den PCs einrichten.

Dann werden wir gemeinsam mit der Entwicklerumgebung Code::Blocks (kann kostenlos von www.codeblocks.org für Windows, Linux und Mac heruntergeladen werden) ein kleines Projekt erstellen, dieses compilieren, linken und starten. Ferner werden wir das Debuggen ausprobieren: Setzen von Breakpoints, schrittweise Ausführung des Programms, Anzeigen von Variableninhalten (Watches).

Im zweiten Teil der Übung werden Sie in Dreiergruppen die eigentliche erste Übung bearbeiten:

Schreiben Sie die fehlenden Funktionen zum vorgegebenen Hauptprogramm. Diese Funktionen sollen in einem eigenen Modul (z.B. `dateutils.c`) untergebracht werden, da sie in den weiteren Übungsaufgaben in angepasster Form 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 Tag, Monat und Jahr) ein gültiges Datum ist. Dabei sollen auch die Schaltjahre berücksichtigt werden.

Die Funktion `getDateFromString` (soll von der nächsten Funktion aufgerufen werden) soll als Funktionsergebnis eine ganze Zahl zurückgeben. Diese Zahl soll als Wahrheitswert angeben, ob in der angegebenen Zeichenkette (1. Parameter) ein Datum enthalten ist. Dieses Datum soll in Tag, Monat und Jahr (als Zahlen) zerlegt – sozusagen geparst – werden. Natürlich sollen diese drei Zahlen als Datum geprüft werden, ob sie ein gültiges Datum ergeben. Wenn ja, sollen diese drei Zahlen in den übergebenen `int`-Zeigern (2. bis 4. Parameter) gespeichert werden.

In der Funktion `getDate` soll der Anwender ein Datum eingeben können, bis ein gültiges Datum eingegeben wurde. Die Funktion erhält als Parameter die Eingabeaufforderung, die jeweils auf dem Bildschirm ausgeben werden soll, sowie drei Zeiger auf Zahlen, die Tag, Monat und Jahr des Datums darstellen. Der Anwender soll das Datum zunächst als Zeichenkette eingegeben. Mit dieser Zeichenkette und den drei Zeiger-Parametern soll dann die Funktion `getDateFromString` aufgerufen werden, die die eigentliche Arbeit der Datumserkennung durchführt. Zurückgegeben wird ein Wahrheitswert, ob ein gültiges Datum eingegeben wurde (da dies in einer Schleife solange durchgeführt wird, bis ein gültiges Datum eingegeben wurde, wird folglich immer eine 1 zurückgegeben).

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");`)
- `askAgain()`

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

Die Headerdatei `string.h` darf nicht 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` schrittweise vorgeführt und dabei alle Variableninhalte angezeigt werden.

Die Bildschirmausgabe soll folgendermaßen aussehen:

```
Geben Sie bitte ein gueltiges Datum ein: 31.02.2013
Ungueltiges Datum!
Geben Sie bitte ein gueltiges Datum ein: 29.02.1900
Ungueltiges Datum!
Geben Sie bitte ein gueltiges Datum ein: 29.02.2000
Das Datum 29.02.2000 ist gueltig!
Moechten Sie noch einmal (j/n) ?
```

Quellcode:

```
#include <stdio.h>
#include "dateutils.h"
#include "tools.h"

int main()
{
   int D, M, Y;

   do
   {
      clearScreen();
      if (getDate("Geben Sie bitte ein gueltiges Datum ein: ", &D, &M, &Y))
         printf("Das Datum %02i.%02i.%04i ist gueltig!\n", D, M, Y);
      else
         printf("Das eingegebene Datum ist ungueltig!\n");
   } while (askAgain());

   return 0;
}
```