



Lehrveranstaltung "Informatik I für TI-Bachelor"

Übungsblatt 6

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 **05. Juni 2015**.

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

Aufgabe: Ziel dieser Übung ist das Erstellen von Nassi-Shneiderman-Diagrammen (Struktogramme nach DIN 66261). Näheres zu diesem Thema finden Sie z.B. bei Wikipedia (<http://de.wikipedia.org>).

Erstellen Sie ein Nassi-Shneiderman-Diagramm für das unten stehende Programm (ueb06.c). Dabei sollen vor allem die Kontrollstrukturen dargestellt werden; die einzelnen Anweisungen können zusammen gefasst werden (z.B. reicht es vollkommen aus, eine Anweisung „Menüausgabe“ für die komplette Ausgabe des Menüs auf dem Bildschirm zu schreiben, anstatt lauter einzelne `printf`-Anweisungen zu erstellen). Auch sollen die Anweisungen in Umgangssprache und nicht speziell in C geschrieben werden (z.B. „a gleich 0?“ anstatt „`if (a == 0)`“). Schließlich soll das Programm anhand eines solchen Diagramms in beliebigen Programmiersprachen erstellt werden können.

Zur Erstellung des Struktogrammes am PC gibt es verschiedene Programme, aber empfehlen möchte ich folgendes (zumindest für diese Übung):

Structorizer Version 1.31 für Windows (<http://structorizer.fisch.lu> im Bereich Downloads; es gibt auch Versionen für Java, Linux und Mac OSX, aber diese Windows-Version läuft für unsere Zwecke stabil!).

```

ueb06.c: #include <stdio.h>
#include <math.h>
#include "escapesequenzen.h"
// aus dem Skript "Grundlagen der Informatik", Kap. 6.3

int main()
{
    char Dummy;                // für Tastaturpuffer-Löschen
    int Menuewahl;             // Benutzereingabe Menüauswahl
    int Eingabe;               // Ergebnis von scanf
    double X;                  // Laufvariable
    double X_Von, X_Bis;       // Ausgabegrenzen
    double X_Schrittweite;     // Ausgabeschrittweite

    do
    {
        // Menue ausgeben
        CLEAR;                  // Bildschirm löschen
        HOME;
        printf("Sinus & Co.\n");
        printf("=====\n\n");
        printf("1. sin(x)\n");
        printf("2. cos(x)\n");
        printf("3. tan(x)\n");
        printf("9. Programmende\n\n");

        // Benutzereingabe Menue
        do
        {
            POSITION(9, 1);      // Eingabezeile löschen
            CLEAR_LINE;
            POSITION(9, 1);
            printf("Ihre Wahl: ");
            Eingabe = scanf("%i", &Menuewahl);
            do                    // Tastaturpuffer loeschen
            {
                scanf("%c", &Dummy);
            } while (Dummy != '\n');

            if (Eingabe)         // wenn Zahl eingegeben
            {                    // dann pruefen, ob gueltig
                POSITION(11, 1);
                switch(Menuewahl)
                {
                    case 1: printf("SINUS\n-----\n"); break;
                    case 2: printf("COSINUS\n-----\n"); break;
                    case 3: printf("TANGENS\n-----\n"); break;
                    case 9: printf("PROGRAMMENDE\n\n"); break;
                    default: Eingabe = 0;
                }
            }
        } while (!Eingabe);      // solange bis gueltiger
                                // Menüpunkt gewählt wurde

        if (Menuewahl != 9)     // außer bei Programmende
        {
            // Startwert erfragen
            do
            {
                POSITION(14, 1); // Eingabezeile löschen
                CLEAR_LINE;
                POSITION(14, 1);
                printf("Von x = ");
                Eingabe = scanf("%lf", &X_Von);
            }
        }
    }
}

```

```

        do                                // Tastaturpuffer loeschen
            scanf("%c", &Dummy);
            while (Dummy != '\n');
    } while (!Eingabe);                    // solange bis gültige Eingabe

    // Endwert erfragen
    do
    {
        POSITION(15, 1);                    // Eingabezeile löschen
        CLEAR_LINE;
        POSITION(15, 1);
        printf("Bis x = ");
        Eingabe = scanf("%lf", &X_Bis);
        do                                // Tastaturpuffer loeschen
            scanf("%c", &Dummy);
            while (Dummy != '\n');
        if (Eingabe)
            if (X_Von > X_Bis) // Ausgabegrenzen prüfen
                Eingabe = 0;
    } while (!Eingabe);                    // solange bis gültige Eingabe

    // Schrittweite erfragen
    do
    {
        POSITION(16, 1);                    // Eingabezeile löschen
        CLEAR_LINE;
        POSITION(16, 1);
        printf("Schrittweite = ");
        Eingabe = scanf("%lf", &X_Schrittweite);
        do                                // Tastaturpuffer loeschen
            scanf("%c", &Dummy);
            while (Dummy != '\n');
    } while (!Eingabe);                    // solange bis gültige Eingabe

    // Werte ausgeben
    POSITION(18, 1);
    printf("          X | f(x)\n");
    printf("-----|-----\n");
    for (X = X_Von; X <= X_Bis; X += X_Schrittweite)
    {
        switch(Menuewahl)
        {
            case 1: printf("%f | %f\n", X, sin(X)); break;
            case 2: printf("%f | %f\n", X, cos(X)); break;
            case 3: printf("%f | %f\n", X, tan(X)); break;
        }
    }

    // Eingabetaste abwarten
    printf("\nBitte Eingabetaste druecken ... ");
    do
        scanf("%c", &Dummy);
        while (Dummy != '\n');
    }
} while (Menuewahl != 9);                // solange bis Programmende

return 0;
}

```