
Lehrveranstaltung "Grundlagen der Informatik" Übungsblatt 5

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! 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 **10. November 2023**.

Aufgabe: Ziel dieser Übung ist das Anwenden von `if`-Abfragen und Schleifen.

Erstellen Sie ein Programm, das eine ganze Zahl vom Benutzer einliest und wieder auf dem Bildschirm ausgibt. Am Ende soll der Benutzer entscheiden können, ob er noch einmal möchte oder nicht.

Das klingt erst einmal sehr einfach, aber es sollen diesmal auch falsche Eingaben abgefragt werden. Dazu soll zum Einen das Ergebnis von `scanf` geprüft werden. Dieses Funktionsergebnis liefert die Anzahl der eingelesenen Variablen. Wird also eine Zahl eingegeben, liefert `scanf` eine `1` zurück. Kann `scanf` die Eingabe aber nicht als Zahl interpretieren, liefert die Funktion eine `0` zurück.

Zum Anderen muss nach jeder Eingabe mit `scanf` der Tastaturpuffer geleert werden. Denn nach dem Einlesen der Zahl bleibt noch mindestens die Eingabetaste (`'\n'`) im Tastaturpuffer stehen, während beim Einlesen eines Zeichens nur dann noch Zeichen im Tastaturpuffer stehen, wenn mehr als nur die Eingabetaste eingegeben wurde. Wie das Beispiel unten zeigt, reicht es aber nicht aus, nur ein Zeichen aus dem Tastaturpuffer auszulesen. Es muss also in einer Schleife solange zeichenweise gelesen werden, bis ein `'\n'` gelesen wurde.

Das Programm soll benutzerfreundlich sein, d.h. dem Benutzer muss mitgeteilt werden, was er tun soll und was er falsch gemacht hat.

Das Compilieren, Linken und Starten des Programms soll wieder mittels einer Make-Datei durchgeführt werden.

Beispiele: *(Die Eingaben sind grau hinterlegt.)*

Geben Sie bitte eine ganze Zahl ein: 123Test

Sie haben die ganze Zahl 123 eingegeben.

Moechten Sie noch einmal (j/n)? ja

Geben Sie bitte eine ganze Zahl ein: Test123

Falsche Eingabe! Die Eingabe ist keine ganze Zahl!

Moechten Sie noch einmal (j/n)? y

Falsche Eingabe! Bitte 'j' oder 'n' eingeben!

Moechten Sie noch einmal (j/n)? J

Geben Sie bitte eine ganze Zahl ein: 987.654

Sie haben die ganze Zahl 987 eingegeben.

Moechten Sie noch einmal (j/n)? nö

Programm wird beendet.