FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Einführung in die Programmierung mit C – WS24/25

7. Übung

Funktionen

Biemann Nikbakhsh Gibietz

16. Dezember 2024

Hinweis: Für das gesamte Semester gilt: Benutzen Sie *keine* C++ Referenzparameter! Beispiel wie es *nicht* sein sollte: `myfct(type & var)`

Aufgabe 1

a) Programmieren Sie eine Funktion `eingabeaufforderung_info`, welche folgenden Text auf dem Bildschirm ausgibt:

```
Es ist nun eine Benutzereingabe erforderlich.
```

b) Programmieren Sie eine Funktion `eingabeaufforderung_zk`, welche folgenden Text auf dem Bildschirm ausgibt:

```
Es ist nun eine Benutzereingabe erforderlich.
Bitte geben Sie eine Zeichenkette ein:
```

Benutzen Sie dazu bereits vorhandene Funktionen und rufen Sie diese gegebenenfalls auf.

c) Programmieren Sie eine Funktion `eingabeaufforderung_int`, welche folgenden Text auf dem Bildschirm ausgibt:

```
Es ist nun eine Benutzereingabe erforderlich.
Bitte geben Sie eine ganze Zahl ein:
```

Benutzen Sie dazu bereits vorhandene Funktionen und rufen Sie diese gegebenenfalls auf.

d) Schreiben Sie ein Hauptprogramm, welches den Benutzer dazu auffordert, eine Zahl einzugeben. Der Benutzer soll solange zur Eingabe einer Zahl aufgefordert werden bis die Zahl positiv ist. Im weiteren Verlauf soll der Benutzer dazu aufgefordert werden, eine Zeichenkette einzugeben. Der Benutzer soll solange zur Eingabe einer Zeichenkette aufgefordert werden bis dieser etwas eingibt; das bedeutet, leere Zeichenketten sollen nicht akzeptiert werden. Beispiel:

```
Es ist nun eine Benutzereingabe erforderlich.
Bitte geben Sie eine ganze Zahl ein: -7
Es ist nun eine Benutzereingabe erforderlich.
Bitte geben Sie eine ganze Zahl ein: -17
Es ist nun eine Benutzereingabe erforderlich.
Bitte geben Sie eine ganze Zahl ein: 5
Es ist nun eine Benutzereingabe erforderlich.
Bitte geben Sie eine Zeichenkette ein:
Es ist nun eine Benutzereingabe erforderlich.
Bitte geben Sie eine Zeichenkette ein: Hallo
```

Benutzen Sie dazu bereits vorhandene Funktionen und rufen Sie diese gegebenenfalls auf.

e) Schreiben Sie nun eine Funktion `ausgabe_int`, welche als Parameter eine ganze Zahl übergeben bekommt. Diese Zahl soll mit einem Kommentar auf dem Bildschirm ausgegeben werden. Beispiel:

```
Die Zahl lautet: 5
```

Erweitern Sie das Hauptprogramm dahingehend, dass die eingegebene Zahl mit Hilfe dieser Funktion auf dem Bildschirm ausgegeben wird.

f) Schreiben Sie nun eine Funktion `ausgabe_zk`, welche als Parameter eine Zeichenkette übergeben bekommt. Diese Zeichenkette soll zwischen zwei Anführungszeichen mit einem Kommentar auf dem Bildschirm ausgegeben werden. Beispiel:

```
Die Zeichenkette lautet "Hallo".
```

Erweitern Sie das Hauptprogramm dahingehend, dass die eingegebene Zeichenkette mit Hilfe dieser Funktion auf dem Bildschirm ausgegeben wird.

Es gibt zwei Möglichkeiten, den für diese Funktion notwendigen Parameter zu deklarieren. Zeigen Sie beide Möglichkeiten! Gibt es Vor- oder Nachteile der jeweiligen Methode?

Aufgabe 2

Schreiben Sie ein Programm zum Einlesen von zwei Zahlen und Ausgabe der Summe dieser beiden Zahlen.

- Die Benutzereingaben sollen jeweils mit passenden Eingabeaufforderungen erfolgen.
- Die Ausgabe soll ebenfalls mit passendem Kommentar erfolgen.
- Lagern Sie möglichst viel redundanten Quelltext in Funktionen aus. Überlegen Sie sich jeweils, welche Parameter und welche Rückgabewerte sinnvoll sind.
- Folgende Vorgaben gelten ebenfalls:
  - Es gibt eine Funktion zum Einlesen einer einzigen Zahl. Die eingelesene Zahl wird als Rückgabewert zurückgeliefert.
  - Es gibt eine Funktion zum Einlesen zweier ganzer Zahlen, welche die Funktion zum Einlesen einer einzigen Zahl aufruft.
  - Es gibt eine Funktion zur Ausgabe zweier Zahlen.
  - Es gibt eine Funktion zur Berechnung der Summe zweier Zahlen.
  - Es gibt eine Funktion zur Ausgabe der Summe zweier Zahlen.

Aufgabe 3

Betrachten Sie folgenden Quelltext:

```
1  int i = 0;
2  printf("i=%d\n", i++);  // Ausgabe: i=0
3  printf("i=%d\n", i++);  // Ausgabe: i=1
4  printf("i=%d\n", i++);  // Ausgabe: i=2
```

Schreiben Sie nun eine Funktion mit dem Bezeichner `iplusplus`, welche die Funktionsweise von `i++` nachahmt. Dabei sei die Bildschirmausgabe für den folgenden Quell-

text identisch mit dem vorherigen.

```
1  int i = 0;
2  printf("i=%d\n", … );  // Ausgabe: i=0
3  printf("i=%d\n", … );  // Ausgabe: i=1
4  printf("i=%d\n", … );  // Ausgabe: i=2
```

Aufgabe 4

Programmieren Sie eine neue Variante der Münzgeldrückgabe unter Zuhilfenahme von Schleifen und Funktionen. Benutzen Sie für die Funktionen *ausschließlich Werteparameter und keine Pointer/Referenzparameter*. Nutzen Sie Schleifen und Funktionen dahingehend, um Ihren Quelltext kompakter zu gestalten. Weiterhin soll keine Limitierung über einen Höchstbetrag erfolgen.

Ein möglicher Programmablauf sei wie folgt:

```
1  Wie groß ist der Euro-Betrag, welcher in moeglichst wenig
      Muenzen aufgeteilt werden soll? 88.88
2  44 x 2-Euro-Muenze
3  0 x 1-Euro-Muenze
4  1 x 50-Cent-Muenze
5  1 x 20-Cent-Muenze
6  1 x 10-Cent-Muenze
7  1 x 5-Cent-Muenze
8  1 x 2-Cent-Muenze
9  1 x 1-Cent-Muenze
```

Aufgabe 5

Erstellen Sie eine weitere Variante der Münzgeldrückgabe. Dieses Mal können Sie neben klassischen Werteparametern auch Pointer übergeben. Nutzen Sie dies, um den Quelltext des Programms deutlich zu reduzieren.

Zur Erinnerung: Für das gesamte Semester gilt: Benutzen Sie *keine* C++ Referenzparameter! Beispiel wie es *nicht* sein sollte: `myfct(type & var)`

Aufgabe 6

Schreiben Sie ein Programm zum Berechnen des Minimums, des Maximums und des Durchschnitts von Zahlen. Dabei soll der Benutzer die Zahlen eingeben, das Programm

Minimum, Maximum und Durchschnitt berechnen und mit geeigneten Kommentaren auf dem Bildschirm ausgeben.

Lagern Sie dabei folgende Funktionalitäten in Funktionen aus:

- Das Einlesen der Zahlen
- Die Berechnung von Minimum, Maximum und Durchschnitt
- Das Ausgeben der Ergebnisse
- Gerne auch mehr

a) Programmieren Sie es für exakt 10 Zahlen.

b) Programmieren Sie es für beliebig viele Zahlen entweder

- in der leichten Variante, in welcher zuerst gefragt wird wie viele Zahlen eingelesen werden sollen, oder
- in der etwas schwereren Variante, in welcher erst am Ende der Eingabe klar ist wie viele Zahlen eingelesen werden; z. B. durch die Abfrage in jedem Schritt, ob der Benutzer noch eine Zahl eingeben will.