SYSTEMPROGRAMMIERUNG: 02. Aufgabenblatt

WS 2013/2014
Prof. Dr.-Ing. C. Gips, Dipl.-Inf. Birgit C. George

FH Bielefeld
University of
Applied Sciences

Funktionen, Strukturen

Ausgabe: 17.10.2013
Abgabe: KW43 (22./25.10.2013)
Punkte: 14+0

1. Aufgabe: Personalverwaltung — 6 Punkte

Schreiben Sie ein (ANSI-) C-Programm für eine einfache Personalverwaltung.

1. Erstellen Sie einen Aufzählungstyp **Bereich**, der die Unternehmensbereiche Management, Entwicklung, Produktion, Marketing und Vertrieb umfasst.
2. Definieren Sie einen neuen Datentypen **Angestellter**, der folgende Informationen enthält:
   - **name** (Char-Array fester Länge)
   - **vorname** (Char-Array fester Länge)
   - **personalNummer** (positive ganze Zahl)
   - **abteilung** (vom Typ Bereich)
   - **gehalt** (reelle Zahl, einfache Genauigkeit)
3. Definieren Sie global ein Feld, welches 10 Angestellte umfassen kann und welches Sie mit mindestens 4 verschiedenen Angestellen initialisieren.
4. Definieren Sie eine parameterlose Funktion, die zu dem globalen Feld aus dem vorigen Aufgabenteil das Durchschnittsgehalt über alle Angestellten berechnet und als Ergebnis zurück liefert.
5. Definieren Sie eine weitere parameterlose Funktion, die für alle Angestellten (im globalen Feld aus dem vorigen Aufgabenteil gehalten) eine Gehaltserhöhung durchführt. Das Management soll 30%, die Entwickler 10%, Produktion und Vertrieb sollen jeweils 1% und das Marketing 2% Gehaltserhöhung erhalten. Verwenden Sie hierfür keine `if`- oder `switch`-Anweisung! (*Hinweis*: Überlegen Sie, was Sie mit dem im ersten Teil definierten Aufzählungstypen anstellen können.)
6. Definieren Sie eine Funktion, mit der Sie **einen** Angestellten vernünftig formatiert ausgeben können. Nutzen Sie die Funktion `printf` aus `stdio.h`. Schreiben Sie eine weitere Funktion, die das globale Feld mit allen Angestellten ausgibt unter Zuhilfenahme der eben implementierten Funktion für die Ausgabe eines Angestellten.
7. Rufen Sie in Ihrer `main`-Funktion die Funktionen für die Berechnung des Durchschnittsgehalts sowie die für die Gehaltserhöhung auf und geben Sie das Ergebnis jeweils vernünftig formatiert aus. Nutzen Sie für die Ausgabe des Feldes Ihre Ausgabefunktion aus dem vorigen Aufgabenteil.

Ziel: Umgang mit Basisdatentypen und Strukturen, Umgang mit Funktionen

2. Aufgabe: Ein- und Ausgabe — 3 Punkte

Schreiben Sie ein (ANSI-) C-Programm, welches für die Eingabe einer positiven ganzen Zahl n folgende Ausgaben produziert. Nutzen Sie dazu die Funktionen `scanf` und `printf` aus `stdio.h`. Nutzen Sie unterschiedliche Schleifenkonstrukte!

1. `***...***`

   (n Sterne in einer Reihe)
2. ```
   *
   **
   ***
   ...
   ***...***
   ```

   (linksbündige Pyramide, beginnend mit einem Stern, je Zeile ein Stern mehr, in Zeile n dann Reihe mit n Sternen)
   ```

3.
```
      *
     ***
     ...
***...***
```
(zentrierte Pyramide, beginnend mit einem Stern, in Zeile i entsprechend $2i-1$ Sterne, letzte Reihe mit n Sternen; n muss ungerade sein!)

Ziel: Ein-/Ausgabe, Schleifen (`for`, `while`)

3. Aufgabe: Casts — 2 Punkte

Geben Sie alle unnötigen bzw. unsinnigen Typumwandlungen (casts) in folgendem Codefragment an und erklären Sie, warum die Umwandlung sinnvoll ist oder nicht:

```
double x;
x = (float)7/4;                  x = (double)(7/4);
x = (double)(7/(float)4);        x = (double)7/4;
x = (double)(7/4.0);             x = (double)7.0f/4;
```

Ziel: implizite und explizite Typumwandlungen

4. Aufgabe: Rätsel? — 3 Punkte

Betrachten Sie folgende Code-Ausschnitte. Erklären Sie, ob sich dort in Bezug auf (ANSI-) C ein Fehler versteckt und falls ja, wie dieser zu beheben wäre. Was ist das Ergebnis folgender Programme?
Begründen Sie Ihre Antwort.

```c
/* Programm A */
double cube(int);
int cube(int);
```

```c
/* Programm B */
double square(double number) {
    double number;
    return number*number;
}
```

```c
/* Programm C */
float cube();    // function prototype
...
cube(float number) {  // function definition
    return number*number*number;
}
```

Ziel: C-Semantik, „Falsche Freunde“ in C