

Prinzip¹

Default-Parameter dienen dazu, beim Aufruf einer Funktion nicht alle Parameter explizit angeben zu müssen. Die nicht angegebenen Parameter werden mit einer Voreinstellung belegt.

Die Funktion im folgenden Beispiel berechnet die Summe von zwei, drei oder vier int-Werten.

```
int summe(int a, int b, int c = 0, int d = 0); // Deklaration

int main()
{
    int x = summe(2, 3, 4, 5); // Ergebnis 14
    x = summe(2, 3, 4);        // Ergebnis 9, es wird d=0 gesetzt
    x = summe(2, 3);           // Ergebnis 5, es wird c=0, d=0 gesetzt
}

int summe(int a, int b, int c, int d) // Definition
{
    return a + b + c + d;
}
```

Default-Parameter werden in der Deklaration einer Funktion angegeben, da der Compiler sie beim Aufruf der Funktion kennen muss. Bei der Funktionsdefinition darf man sie nicht schreiben, außer, sie steht über dem Hauptprogramm.

Anordnung der Default-Werte

Default-Parameter müssen von rechts her angegeben werden und es darf keine Lücke mit einem normalen Parameter dazwischen liegen.

Beispiel für eine **ungültige** Funktionsdeklaration:

```
int summe(int a, int b = 0, int c = 0, int d); // Wert für d fehlt!
```

¹ Dieser Abschnitt ist eine bearbeitete Version des Kapitels *Default-Parameter* aus dem Wikibook zu C++ (http://de.wikibooks.org/wiki/C%2B%2B-Programmierung/_Weitere_Grundelemente/_Prozeduren_und_Funktionen, Stand 04.11.2013, CC-BY-SA 3.0).

Aufgaben

Aufgabe 1: Füllzeichen 2

Ergänzen Sie Ihre Funktion aus der Aufgabe „Füllzeichen“ so, dass 10 Zeichen ausgegeben werden, wenn man beim Aufruf keine Anzahl übergibt.

Aufgabe 2: Einmaleins

Schreiben Sie eine Funktion mit dem Prototyp `void einmaleins(unsigned ende=10, unsigned faktor=5)`, die eine Einmaleins-Tabelle für Grundschüler auf dem Bildschirm ausgibt.

Ausgabe im Default-Fall:

```
1 * 5 = 5
2 * 5 = 10
3 * 5 = 15
4 * 5 = 20
5 * 5 = 25
6 * 5 = 30
7 * 5 = 35
8 * 5 = 40
9 * 5 = 45
10 * 5 = 50
```

Die Tabelle beginnt immer bei `1*` . . . und endet beim Wert des ersten Parameters, im Default-Fall `10*` Der zweite Parameter ist der Faktor, mit dem multipliziert wird.

Die Funktion soll auf unsinnige oder sehr große Parameterwerte mit einer passenden Fehlermeldung reagieren.

Expertenaufgabe: Runden

Schreiben Sie eine Funktion mit dem folgenden Prototyp:

```
double runden(double wert, unsigned stellen=2);
```

Das Unterprogramm soll den übergebenen Wert auf die angegebene Zahl von Nachkommastellen runden, ohne eine Bibliotheksfunktion zu benutzen. Bedenken Sie, dass auch negative Werte vorkommen können.