

Einführung in die Programmierung mit C++

Übungsblatt 4

Kontrollfluss und Datentypen

Sebastian Christodoulou Alexander Fleming Uwe Naumann

Informatik 12:
Software and Tools for Computational Engineering (STCE)
RWTH Aachen

In `main1.cpp` wird eine typgenerische Funktion implementiert.

1. Schreibe ein Programm, das die folgende Formel mit Hilfe einer Funktion `f` berechnet:

$$f(a, b) = (a/b) * (b/a) .$$

Die Funktion soll typgenerisch implementiert werden, so dass sich die Datentypen der Parameter `a` und `b` unterscheiden dürfen. Der Einfachheit halber gibt die Funktion immer eine Variable vom Typ **double**, zurück.

2. Erstelle in der Hauptroutine zwei **int**-Variablen `i1` und `i2` sowie eine **double**-Variable `d`. Wähle die Werte paarweise verschieden und ungleich 0.
3. Rufe die Funktion `f` für die Kombinationen `(i1,i1)`, `(i1,i2)`, `(i1,d)`, `(d,i1)`, `(d,d)` auf und gebe die Resultate auf der Kommandozeile aus.
4. Sind die Resultate wie erwartet? Falls nein, welche sind *mathematisch* falsch und woran könnte das liegen? Schreibe deine Antwort in die Kommentare zu den aufrufen

In `main2.cpp` werden Dezimalzahlen in Binärdarstellung umgewandelt.

1. Bestimme die Binärdarstellung der Dezimalzahlen 69 und -31 mit 8 Bits. Erläutere deine Rechnungen kurz in den Kommentaren.
2. Schreibe ein Programm, das eine positive ganze Zahl z über eine Kommandozeilen-Eingabe (`cin`) übergeben bekommt und die Binärdarstellung dieser Zahl auf der Kommandozeile ausgibt. Die Anzahl der Bits n , die benötigt werden um die Zahl z darzustellen kann mit

$$n = \left\lceil \frac{\log(z + 1)}{\log(2)} \right\rceil ,$$

berechnet werden. Die Klammern bedeuten, dass das Ergebnis zur nächsten ganzen Zahl aufgerundet wird. Aufrunden ist durch die Funktion `ceil` aus dem `<cmath>`-Teil der Standardbibliothek implementiert. Nutze eine Schleife um über die Bits zu iterieren und zu bestimmen, ob ein Bit 0 oder 1 ist. Ein möglicher Programmablauf könnte wie folgt aussehen:

Eingabe Zahl: > 11

Ausgabe: 1011

In `main3.cpp` wird die Binärdarstellung positiver Zahlen als Dezimalzahl interpretiert.

1. Für eine eingegebene positive Binärzahl, wird der Nutzer zuerst gefragt wie viele Bits die Binärzahl hat. Anschließend sollen diese nacheinander abgefragt werden. Ein möglicher Programmablauf könnte wie folgt aussehen:

```
Anzahl der Bits:  > 4
1.  Bit:  > 1
2.  Bit:  > 0
3.  Bit:  > 1
4.  Bit:  > 1
Ausgabe:  11
```

Abgaben

- ▶ main1.cpp
- ▶ main2.cpp
- ▶ main3.cpp