

## Serie 2

### *Vertiefung der Programmierung mit C++*

## Übungsaufgaben

### **Aufgabe 1 – C++ lesen und verstehen**

Erläutern Sie die nummerierten Zeilen des folgenden Programms! Welche Ausgaben werden durch das Programm erzeugt?

```
01 #include <iostream>
    #include <fstream>
    #include <cassert>

02 using namespace std;

03 namespace inc {
04     int f(int x) {
04+         return ++x;
        }
    }

05 void f(int& x, int amount=2) {
06     assert(x >= amount);
06+     x -= amount;
    }

    int divide(int a, int b) {
07     if (b==0) throw "Divide by zero!";
        return a / b;
    }

    int main() {
        int x;

08     x = inc::f(1);
08+     f(x);

09     cout << (x == 1 ? "1" : "0");
        cout << endl;

        try {
10             ofstream f;
10+             f.open("out.txt");
11             if (f.good())
11+                 f << divide(10, x) << endl;
                f.close();

                int i = 3;
                int a[3] = { 2, 4, 8 };
                do {
12+                     a[--i]++;
                } while (i);
                cout << a[0] << endl;
            }

13     catch (const char* ex) {
13+         cerr << "Error: " << ex << endl;
            }
        }
```

### Aufgabe 2 – Iteration und Rekursion

Zur Berechnung der Potenz  $x^y$  zweier natürlichen Zahlen  $x$  und  $y$  ist wiederholtes Multiplizieren notwendig! Schreiben Sie zwei Funktionen, die die Potenz  $x^y$  berechnen. Eine Funktion soll das wiederholte Multiplizieren durch Iteration implementieren, die andere Funktion durch Rekursion. Testen Sie beide Lösungen mit Hilfe eines geeigneten Hauptprogramms.

### Aufgabe 3 – Pascalsches Dreieck ausgeben mit C++

Das Pascalsche Dreieck ist eine Form der grafischen Darstellung der Binomialkoeffizienten, die auch eine einfache Berechnung dieser erlaubt.

$$\begin{array}{ccccc}
 \binom{0}{0} & \binom{1}{0} & \binom{2}{0} & \binom{3}{0} & \binom{4}{0} \\
 \binom{1}{1} & \binom{2}{1} & \binom{3}{1} & \binom{4}{1} & \\
 \binom{2}{2} & \binom{3}{2} & \binom{4}{2} & & \\
 \binom{3}{3} & \binom{4}{3} & & & \\
 \binom{4}{4} & & & & 
 \end{array}
 \rightarrow
 \begin{array}{ccccc}
 1 & 1 & 1 & 1 & 1 \\
 1 & 2 & 3 & 4 & \\
 1 & 3 & 6 & & \\
 1 & 4 & & & \\
 1 & & & & 
 \end{array}$$



Die Binomialkoeffizienten sind im Dreieck derart angeordnet, dass jeder Eintrag  $b_{i,j}$  die Summe zweier vorheriger Einträge ist. Dabei kann die Variable  $i$  als Zeilenindex und  $j$  als Spaltenindex interpretiert werden, wobei die Zählung mit Null beginnt (also erste Zeile  $i=0$ , erste Spalte  $j=0$ ). Beginnt man in der Zeile  $i=0$  und der Spalte  $j=0$  mit dem Wert 1, so ergeben sich für  $i > 0$  und  $j > 0$  die Binomialkoeffizienten durch  $b_{i,j} = b_{i-1,j} + b_{i,j-1}$ .

Schreiben Sie ein iteratives C++ Programm zur Ausgabe des Pascalschen Dreiecks der Größe  $n$  ( $n < 20$ )! Die Größe  $n$  ist von der Tastatur einzulesen. Als mathematische Operationen sei **nur die Addition** zugelassen, d.h., es soll **keine Multiplikation** oder **Division** stattfinden.

## Hausaufgaben

### Aufgabe 1 – Lösen quadratischer Gleichungen (Punkte: 7)

Entwickeln Sie ein Programm zum Lösen quadratischer Gleichungen der Form  $ax^2+bx+c=0$ . Berechnen Sie nur reelle Lösungen! Folgende Teilaufgaben sind dafür zu bearbeiten:

- Daten (1 Punkt)** Definieren Sie eine Struktur `struct QuadEquation {...}`, welche sowohl die Koeffizienten  $a, b, c$  der quadratischen Gleichung als auch die Lösungen und die Anzahl der Lösungen speichern soll!
- Berechnung (3 Punkte)** Erstellen Sie eine Funktion `.. solveQuad(..)`, welche die Lösung einer quadratischen Gleichung berechnet! Als Parameter wird eine wie in a) definierte Struktur als Referenz übergeben. Die Funktion ermittelt die Anzahl der Lösungen der Gleichung, berechnet diese und speichert die Ergebnisse in der Struktur.
- Ausgabe (1 Punkt)** Schreiben Sie eine Funktion `.. printQuad(..)`, welche eine quadratische Gleichung mit ihren Lösungen auf dem Bildschirm ausgibt! Die Funktion erhält als einzigen Parameter eine Referenz auf eine Konstante `QuadEquation`.
- Testen (2 Punkte)** Testen Sie die Funktionen zum Lösen und Ausgeben quadratischer Gleichungen in einem Hauptprogramm! Lesen Sie die Koeffizienten  $a, b, c$  ein, berechnen Sie die möglichen Lösungen und geben Sie das Ergebnis aus!

### Aufgabe 2 – Wahrheitswertetabellen ausgeben (Punkte: 3)

Schreiben Sie ein Programm zur Berechnung und Ausgabe von Wahrheitswertetabellen von booleschen Formeln. Verwenden Sie Schleifen und Boolesche Operatoren (`!`, `&&`, `||`), um die Variablenbelegungen zu generieren und die entsprechenden Formeln zu berechnen.

Ihr Programm soll die Wahrheitswertetabellen für die folgenden Formeln berechnen und ausgeben:

- NOT  $x$  (einfache Schleife)
- $x$  XOR  $y$  (zwei geschachtelte Schleifen)
- $(x$  NAND  $y)$  NAND  $z$  (drei geschachtelte Schleifen)

Folgendes Beispiel zeigt die Wahrheitswertetabelle des AND Operators.

x	y	x AND y
0	0	0
0	1	0
1	0	0
1	1	1