



Ausgabe: 2008-11-11

Abgabe: 2008-11-18

Objektorientierte Programmierung

6. Übungsblatt

Aufgabe 1 (Punkte: 12)

Führen Sie die folgenden Befehlsfolgen mit Papier, Bleistift und Radiergummi aus und geben Sie genau an, welche Zeichen und Zeilen zur Standardausgabe (d.h. zum Bildschirm) ausgegeben werden.

```
1 // --- 1. Die anna-Schleife: ---
2 int anna = 7;
3 while (anna > 0) {
4     cout << anna << " ";
5     anna /= 2;
6 }
7 cout << endl;;
8
9 // --- 2. Die berta-Schleife: ---
10 int berta = 7;
11 do {
12     berta /= 2;
13     cout << " -> " << berta;
14 } while (berta > 0);
15 cout << endl;;
16
17 // --- 3. Die celia-Schleife: ---
18 for (int celia = -3; celia < 5; celia += 2) {
19     cout << 2 * celia + 4 << " ";
20 }
21 cout << endl;;
22
23 // --- 4. Die dora-Schleife: ---
24 for (int dora = 3; 2*dora > -3; dora--) {
25     cout << dora << " ";
26 }
27 cout << endl;;
28
```

```

29 // --- 5. Die MÁX1-Schleife: ---
30 int MAX1 = 3;
31 for (int i1 = 1; i1 <= MAX1; i1++) {
32     for (int i2 = 1; i2 <= 2*MAX1; i2++) {
33         cout << "*";
34     }
35     cout << endl;;
36 }
37
38 // --- 6. Die MAX2-Schleife: ---
39 int MAX2 = 5;
40 for (int i1 = 1; i1 <= MAX2; i1++) {
41     for (int i2 = 1; i2 <= i1; i2++) {
42         cout << "++";
43     }
44     cout << endl;;
45 }

```

Aufgabe 2 (Punkte: 7)

Schreiben Sie ein C++ Programm, welches einen einfachen Taschenrechner simuliert. Gehen Sie dabei folgendermaßen vor: Lesen Sie als erstes den arithmetischen Operator ein. Lesen Sie dann die benötigten Operanden in Variablen vom Typ `float` ein. Sorgen Sie dafür, dass Ihr Programm bei einer Division durch 0 nicht abstürzt. Das Programm sollte diesen Fall vor der Berechnung prüfen, gegebenenfalls eine Fehlermeldung ausgeben und die Programmausführung beenden.

Als Operatoren sind + (Addition), - (Subtraktion), * (Multiplikation) und / (Division) zulässig. Verwenden Sie zur Ablaufsteuerung ein `switch`-Statement.

Geben Sie das Ergebnis der Berechnung in folgender Form aus:

Operand1 Operator Operand2 = Ergebnis