

Marc Gillé Informatik II, OH14/R.312; Tel.: 7723

Amer Krivošija Informatik II, OH14/R.309; Tel.: 4762

Christian Pölitz Informatik VIII, JvF23/R.1.37; Tel.: 8256

Dimitri Scheftelowitsch Informatik IV, MSW18/R.117; Tel.: 5855

Dortmund, den 07. November 2013

Übungen zur Vorlesung EidP (WS 2013)

Blatt 3

Block gelb

Es können 4 Punkte erreicht werden.

Abgabedatum: 14. November 2013 23:59 Uhr

Hinweise

1) Bitte beachten Sie die aktuellen Hinweise unter

<http://ls11-www.cs.tu-dortmund.de/teaching/ep1314uebung/>

- 2) Für die Abgabe sind die Dateien: *Aufgabe_03_0.txt*, *Aufgabe_03_1a.cpp*, *Aufgabe_03_1b.cpp* und *Aufgabe_03_1c.cpp* zu erstellen.
- 3) In der optionalen Datei *Anmerkungen.txt* können Sie allgemeine Anmerkungen bezüglich Ihrer Lösungen notieren.
- 4) Es ist ratsam, die Programme vor der Abgabe zu kompilieren und auszuführen.
- 5) Verwenden Sie für die Textaufgaben reine Texteditoren. Das Abgabesystem erkennt **keine** Word- oder PDF-Dateien!!!

Aufgaben

Aufgabe 0: Grundlagen (1 Punkt)

Speichern Sie Ihre Ergebnisse in der Ergebnisdatei *Aufgabe_03_0.txt*.

- Erklären Sie den Unterschied zwischen einer `for`-Schleife und einer `while`-Schleife (0,2 Punkte).
- Erklären Sie wie sowohl in einer `for`-Schleife als auch in einer `while`-Schleife Endlosschleifen entstehen können (0,2 Punkte).
- Erklären Sie den Unterschied zwischen einer `break`- und einer `continue`-Anweisung (0,2 Punkte).
- Was ist ein "Zeiger" (0,2 Punkte)?
- Welche Adresse referenziert der Zeiger `double *pVar` nachdem diese Deklaration ausgeführt wurde (0,2 Punkte)?

Aufgabe 1: Schleifen (3 Punkte)

Legen Sie für Ihre Antworten die Text-Dateien *Aufgabe_03_1a.cpp*, *Aufgabe_03_1b.cpp* und *Aufgabe_03_1c.cpp* an.

- Eine natürliche Zahl $p > 1$ heißt *Primzahl*, falls p nur durch 1 und durch sich selber ohne Rest teilbar ist. Legen Sie eine Datei *Aufgabe_03_1a.cpp* an und ergänzen Sie darin das folgende C++-Programmfragment so, dass bei Ausführung des Programms die ersten n Primzahlen ausgegeben werden (1 Punkt).

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     unsigned int n = 20;
6
7     //*****
8     // Hier beliebig viele Zeilen ergaenzen
9     //*****
10
11     return 0;
12 }
```

Kompilieren Sie das Programm und führen Sie es anschließend aus. Kopieren Sie die Ergebnisse als Block-Kommentar an das Ende der Datei *Aufgabe_03_1a.cpp*.

Hinweis: Das entstehende Programm soll auch bei „beliebiger“ Veränderung des Wertes von n die korrekten Ausgaben berechnen.

b)

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int i = 666;
6      while (i >= 0) {
7          cout << i << endl;
8          i -= 3;
9      }
10     return 0;
11 }
```

Ändern Sie obiges Programm derart, dass statt einer while-Schleife eine for-Schleife mit vollständigem Schleifenkopf verwendet wird. Schreiben Sie Ihr komplettes Programm in die Datei *Aufgabe_03_1b.cpp*. Kompilieren Sie das Programm und führen Sie es anschließend aus. Kopieren Sie die Ergebnisse als Block-Kommentar an das Ende der Datei *Aufgabe_03_1b.cpp* (1 Punkt).

c) Ändern Sie das Programm aus Teilaufgabe b) derart, dass statt einer while-Schleife eine do-while-Schleife verwendet wird. Schreiben Sie Ihr komplettes Programm in die Datei *Aufgabe_03_1c.cpp*. Kompilieren Sie das Programm und führen Sie es anschließend aus. Kopieren Sie die Ergebnisse als Block-Kommentar an das Ende der Datei *Aufgabe_03_1c.cpp* (1 Punkt).

(Präsenz-)Aufgabe 2: Zeiger (0 Punkte)

Gegeben sei das folgende C++-Programm:

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      short a = 15;
6      short b[] = {122, 151, 110};
7      short *x1 = nullptr;
8      short *x2 = nullptr;
9      short **y1 = nullptr;
10     short **y2 = nullptr;
11
12     x1 = &a;
13     x2 = x1;
14     x1 = &b[1];
15     x2 = x1-1;
16     y1 = &x2;
17     *y1 = x1;
18     y2 = y1;
19     x1 = &a;
20     y1 = &x1;
21     *x2 = a**(*y2+1);
22     **y1 = 1;
23     *(x2-1) = **y2+((*y2+*x1))+3;
24     cout << "a:␣" << a << "\n";
25     cout << "b:␣[" << b[0] << ", ";
26     cout << b[1] << ", " << b[2] << "]\n";
27     return 0;
28 }
```

In der untenstehenden Tabelle sind die Werte einzelner Variablen bzw. Ausdrücke (*nach* der Ausführung der jeweils angegebenen Zeile) aufgeführt. Vervollständigen Sie die Tabelle.

Hinweis: Die Einträge sollen möglichst einfach sein, z. B. steht in der Spalte `*x1` des Beispiels die Zahl 15 und nicht der Ausdruck `*&a`.

Zeile	a	b	x1	*x1	x2	*x2	y1	*y1	**y1	y2	*y2	**y2
10	15	[122, 151, 110]	nullptr	–	nullptr	–	nullptr	–	–	nullptr	–	–
12	15	[122, 151, 110]	&a	15	nullptr	–	nullptr	–	–	nullptr	–	–
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮