

Übung zur Vorlesung EidP (WS 2018/19)

Blatt 3

Block gelb

Es können 4 Punkte erreicht werden.

Abgabedatum: 8. November 2018, 23:59 Uhr

Hinweise

- Bitte beachten Sie die aktuellen Hinweise unter

<https://ls11-www.cs.tu-dortmund.de/teaching/ep1819uebung/>

- Für die Abgabe sind die Quellcode-Dateien `Aufgabe_03_1.txt` und `Aufgabe_03_2.cpp` zu erstellen.
- Stellen Sie sicher, dass alle von Ihnen abgegebene Dateien reine Textdateien im UTF-8-Format sind.
- Für die Kompilierung des Programms muss der C++11-Standard aktiviert sein. Dies kann im Referenzcompiler GCC 4.9 durch den Schalter `-std=c++11` sichergestellt werden. Es soll zudem mit die Parameter `-pedantic` und `-Werror` genutzt werden. Der Befehl zum Kompilieren soll somit wie folgt aussehen:

```
g++-4.9 -pedantic -Werror -std=c++11 Aufgabe_03_02.cpp -o Aufgabe_03_02
```

Aufgaben

Aufgabe 1: Grundlagen (1 Punkt)

Legen Sie für Ihre Antworten eine Text-Datei `Aufgabe_03_1.txt` an.

- Erklären Sie den Unterschied zwischen einer `for`-Schleife und einer `while`-Schleife. (0.2 Punkte)
- Erklären Sie anhand eines Beispiels, wie sowohl in einer `for`-Schleife als auch in einer `while`-Schleife Endlosschleifen entstehen können. (0.2 Punkte)
- Was ist ein *Zeiger*? Was ist der Unterschied zwischen den Zeigeroperatoren `&` und `*`? (0.2 Punkte)

d) Es sei die Deklaration `double *pVar;` gegeben. Welche Adresse referenziert dieser Zeiger nach der Deklaration? (0.1 Punkte)

e) Gegeben ist folgendes Programm. Erklären Sie was das Programm ausführlich und beschreiben Sie insbesondere die Deklaration der Variablen. Ändern Sie das Programm so, dass jede Variable in einer eigenen Zeile deklariert wird.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int* x, y, z[1]={};
6      y = 1;
7      x = &y;
8      z[0] = ++*x;
9      cout << z[*x-2] << endl;
10     return 0;
11 }
```

(0.3 Punkte)

Aufgabe 2: Mehrfachauswahl (3 Punkte)

Ergänzen Sie in der Datei `Aufgabe_03_2.cpp` das unten angegebene Programmfragment so, dass für ein eingegebenes Datum der darauffolgende Tag berechnet wird. Dabei soll zuvor geprüft werden, ob das Datum gültig ist. Wird ein Datum vor dem 1.1.1600 oder nach dem 31.12.2600 eingegeben, soll 99.99.9999 ausgegeben werden. Dasselbe soll ausgegeben werden, wenn ein ungültiges Datum eingegeben wird. Verwenden Sie zur Bestimmung der Anzahl an Tagen des eingegebenen Monats eine Mehrfachauswahl (`switch/case`). Achten Sie dabei insbesondere auf Schaltjahre. Alle Jahre, die durch 400 teilbar sind, sind Schaltjahre. Alle anderen Jahre sind genau dann Schaltjahre, wenn sie durch 4, aber nicht durch 100 teilbar sind.

Kompilieren Sie das Programm und führen Sie es anschließend aus. Kopieren Sie die Ergebnisse für die Eingaben 0.11.2016, 11.11.1111, 30.11.2016, 28.02.2016 und 28.02.2017 als Block-Kommentar an das Ende der Datei `Aufgabe_03_2.cpp`.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      unsigned int tag, monat, jahr;
6      cout << "Geben Sie Tag, Monat und Jahr ein" << endl;
7      cin >> tag;
8      cin >> monat;
9      cin >> jahr;
10     //*****
11     // Hier beliebig viele Zeilen ergaenzen
12     //*****
13     cout << "Naechster Tag ist "
14          << tag << "." << monat << "." << jahr << endl;
15     return 0;
16 }
```

Präsenzaufgabe 3: Zeiger (0 Punkte)

Gegeben sei das folgende C++-Programm:

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      short a = 11;
6      short b[] = { 87, 131, 81 };
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     y1 = &x1;
20     x1 = &a;
21     *x2 = a**(*y2+1);
22     **y1 = 1;
23     *(x2-1) = **y2+((*y2**x1))+3;
24
25     cout << "a: " << a << endl;
26     cout << "b: [" << b[0] << ", ";
27     cout << b[1] << ", " << b[2] << "]" << endl;
28
29     return 0;
30 }
```

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

Zeile	a	b	x1	*x1	x2	*x2	y1	*y1	**y1	y2	*y2	**y2
10	11	[87, 131, 81]	nullptr	-	nullptr	-	nullptr	-	-	nullptr	-	-
12	11	[87, 131, 81]	&a	11	nullptr	-	nullptr	-	-	nullptr	-	-
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Hinweis: Die Einträge sollen möglichst einfach sein, z. B. steht in der Spalte *x1 des Beispiels die Zahl 11 und nicht der Ausdruck *&a.