

Themen: C++ (Schleifen; Funktionen; ein- & mehrdimensionale Arrays; Zufallszahlen; Sortierverfahren)

Aufgabe 1: Verständnisfragen zu Grundlagen Cpp

- 1.) Geben Sie ein Beispiel einer Funktion, die Anweisungen enthält, aber keinen Return-Wert liefert.

Die Funktion „main ()“ hat den Returnwert „return 0“.

- 2.) Mit welcher Standardfunktion kann durch wiederholte Aufrufe eine Folge von Zufallszahlen erzeugt werden?

Wie lautet der dazu notwendige Header?

Funktion „rand ()“ mit dem Header „stdlib.h“ bzw. „time.h“.

- 3.) Beim Aufruf einer Standardfunktion benötigt der Compiler



keine weiteren Informationen?



den Prototyp der Funktion?



die Definition der Funktion?

- 4.) Welchen Rückgabetyt hat der Ausdruck `calc (7, 12.9)`, wenn er gemäß dem Prototyp der Funktion `double calc (int n, float x)` deklariert ist?

Der Rückgabetyt lautet „double“.

- 5.) In welchem Verzeichnis sucht der Compiler folgende Header-Datei:
<iostream.h>

Der Compiler sucht die Datei im include-Verzeichnis des Systemverzeichnisses Cpp, das durch den Adressierungspfad bei der Installation eingerichtet wurde.

- 6.) Erklären Sie hierzu den Unterschied, wenn die Header-Datei in Hochkommas angegeben würde: „iostream.h“

Hier wird die Datei zuerst im Verzeichnis gesucht, in dem auch das erstellte Programm liegt.

Aufgabe 2: Funktionen I - Fehlersuche bei Funktionen

Welche Fehler beinhalten die folgenden Prototypen?

1.) `double calculate double x, float y ;`

Korrektur : Klammern () fehlen
`double calculate (double x, float y) ;`

2.) `int auswahl(int a, b) ;`

Korrektur : jede Variable muss eigens mit ihrem Variablentyp deklariert werden
`int auswahl (int a, int b) ;`

Aufgabe 3: Funktionen II - Erstellen von Funktionen

1.) Die Funktion `sum()` soll die Summe von drei `double`-Werten als Return-Ergebnis liefern.

Lösung:

=> [Quelltext „sum\(\)“ anzeigen](#) => [Programm ausführen](#)

=> [Quelltext „Programm“ anzeigen](#)

2.) Die Funktion `ggt()` soll den größten gemeinsamen Teiler zweier Ganzzahlen ermitteln und als Return-Wert ausgeben

Lösung:

=> [Quelltext „ggt\(\)“ anzeigen](#) => [Programm ausführen](#)

=> [Quelltext „Programm“ anzeigen](#)

Aufgabe 4: Freie Syntaxwahl

Erstellen Sie ein Programm, das nach Eingabe einer mindestens vierstelligen Sekundenanzahl die Anzahl der Stunden, Minuten und Sekunden ausgibt.

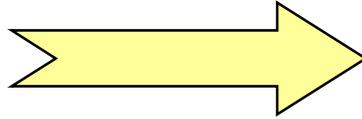
Lösung:

=> [Quelltext anzeigen](#) => [Programm ausführen](#)

Aufgabe 5: Schleifen I

Wandeln Sie die folgende for-Schleife in eine **while**-Schleife um.

```
for (int i = 1 ; i <= n; i++)  
{  
    cout << i*i << " ";  
}
```



```
while (i <= n)  
{  
    cout << i * i << " ";  
    i = i + 1;  
}
```

Aufgabe 6: Quelltextinterpretation

Erläutern Sie kurz den Syntaxausschnitt des folgenden Programms:

```
31 for (versuch=1; versuch<=ende; versuch++)  
32     {  
33     magic = rand() % 2;  
34  
35     summe = magic + summe;  
36  
37     cout <<magic <<"\t";  
38     }  
39  
40 cout <<summe/ende;
```

Lösung:

- Zeile 31: Zählerschleife; Startvariable „versuch“ mit dem Wert 1
- Zeile 33: Erzeugung ganzzahliger Zufallszahlen: 0 oder 1 (Simulation eines Münzwurfs)
- Zeile 35: Addition der Zufallszahlen
- Zeile 40: Ermittlung der relativen Häufigkeit der Zahl 1 (z.B. relative Häufigkeit für Wappen)

Aufgabe 7: Sortieralgorithmen

Hier ist die Syntax eines Ihnen bekannten Sortieralgorithmus gegeben.

- a) Wie heißt der Sortieralgorithmus?
- b) Erklären Sie die Vorgehensweise allgemein und führen Sie das Sortierprinzip anhand folgender 5 Zahlenwerte durch:

12 - 5 - 67 - 23 - 12

- c) Erstellen Sie zu dem gegebenen Sortieralgorithmus das Struktogramm.

Quelltext-Auszug:

```
int i,k,t,n;
for (i=n-1; i>0; i--)
{
    for (k=0; k<i; k++)
    {
        if (daten[k] > daten[k+1])
        {
            t = daten[k];
            daten[k] = daten[k+1];
            daten[k+1] = t;
        }
    }
}
```

Lösung:

- a) Es handelt sich um den Bubblesort.

- b) **Verfahrensbeschreibung:**

Idee: Die zu sortierenden Elemente im Array steigen wie Luftblasen *im Wasser auf*.

- ❶ Durchlaufe den Array in aufsteigender Reihenfolge und betrachte dabei immer zwei benachbarte Elemente.
- ❷ Wenn zwei Elemente in falscher Reihenfolge stehen, dann werden sie vertauscht. Nach diesem Durchlauf ist in jedem Fall das größte Element am Ende des Arrays.
- ❸ Wiederhole den obigen Verfahrensschritt so lange, bis das Array vollständig sortiert ist.

Beispiel: 12 5 67 23 12

(0-1): 12 - 5 (1-2): 12 - 67 (2-3): 67 - 23 (3-4): 67 - 12

1. Durchlauf beendet: 5 12 23 12 67

(0-1): 5 - 12 (1-2): 12 - 23 (2-3): 23 - 12

2. Durchlauf beendet: 5 12 12 23 67

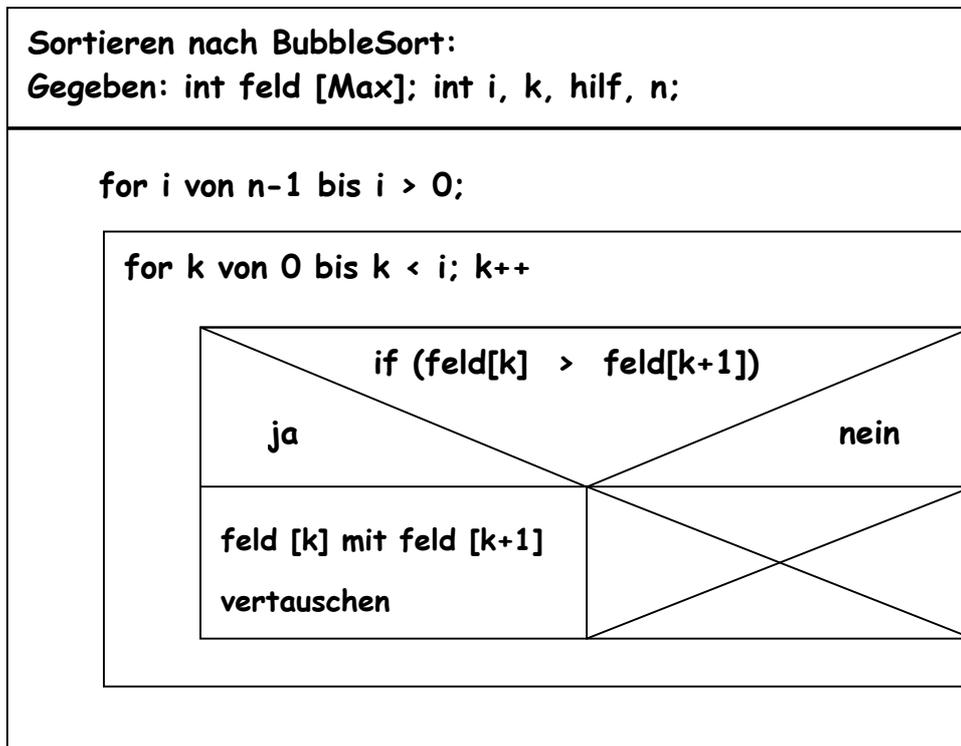
(0-1): 5 - 12 (1-2): 12 - 12

3. Durchlauf beendet: 5 12 12 23 67

(0-1): 5 - 12

4. Durchlauf beendet: 5 12 12 23 67

c)



Aufgabe 8: Ein- und mehrdimensionale Arrays

1.) Definieren Sie den Begriff Array

Unter einem Array versteht man eine Liste bzw. mehrdimensionale Datenstruktur mit Daten des gleichen Variablentyps.

2.) Wie erfolgt die Erzeugung eines Arrays per Initialisierung?

liste[] = { 12,23,4,6,... }

Die folgenden Arbeitsschritte sollen alle in einem einzigen Programm erfasst werden. Bitte kommentieren Sie hierzu alle Teilschritte und Einzelsequenzen im Sinne eines guten Programmierstils aus.

- 3.) Erzeugen Sie ein eindimensionales Array mit 10 Werten aus dem Bereich [1 ; 50] per Zufallsgenerator über eine for-Schleife und lassen sie die Werte anzeigen.

Hinweis: Initialisierung der ZG: `srand((unsigned int)time(NULL));`

- 4.) Lassen Sie dieses Array nun in umgekehrter Reihenfolge ausgeben.
- 5.) Ermitteln Sie das Minimum, das Maximum und den Durchschnittswert des Arrays und lassen Sie die Ergebnisse anzeigen.
- 6.) Erweitern Sie das eindimensionale Array derart, dass Sie nun zu jedem Zahlenwert die Position vergleichbar folgender Tabelle angezeigt bekommen.

Position	0	1	2	3	4	...
Wert	23	3	12	48	37	...

- 7.) Nun soll eine weitere Ergänzung derart erfolgen, dass alle ungeraden Werte, mit einer 0 und alle geraden Werte mit einer 1 dargestellt werden.

Position	0	1	2	3	4	...
Wert	23	3	12	48	37	...
Auswertung	0	0	1	1	0	...

- 8.) Abschließend sollen die Array-Elemente um jeweils zwei Positionen nach rechts verschoben werden, und die beiden letzten Elemente demzufolge am Anfang erscheinen.

Lösung:

=> [Quelltext anzeigen](#)

=> [Programm ausführen](#)

Aufgabe 9: Quellcodesammlung

Übertragen Sie alle Quellcodes der erstellten Programme in eine WORD-Datei geordnet nach den Aufgaben mit jeweiligen Überschriften und Zeilennummern

Lösung: vergleiche die jeweiligen Lösungen => [Quelltexte als pdf](#)