

Lernzettel Test - Programmierung

Themen: Selektion & Schleifen

Jedes Programm beginnt mit:

```
1 #include <iostream>
2 using namespace std;
```

Danach wird hinzugefügt:

```
3
4 int main () {
5     [redacted]
6 }
```

Und hier wird nun unser Code reingeschrieben

Am Ende jeden Codes sollte außerdem folgendes stehen:

```
    return 0;
}
```

(Die ") ist immer noch die gleiche, wie im Bild darüber)

while Schleife (Kopfgesteuert)

Eine while Schleife beginnt immer mit `while` und einer Kondition, welche erfüllt sein muss, in Klammern. Ist die Kondition nicht erfüllt, wird die while Schleife nicht ausgeführt.

```
1 #include <iostream>
2 using namespace std;
3
4 int main () {
5     while (Kondition) {
6         Ausfuehrung
7     }
8 }
```

Eine unendliche while Schleife würde so aussehen:

```
int main() {  
    int zahl = 1;  
    while (zahl = 1) {  
        cout << "Jap, 1 ist immer noch 1.";  
    }  
}
```

Hier wird festgelegt, dass die Variable "Zahl" gleich 1 entspricht. Und solange diese Variable den Wert 1 hat, kriegen wir einen ssatz in die Konsole geprinted.

Dieser angepasste Code löst das Problem, indem der user eine neue Zahl für die Variable eingeben kann. Das Programm wird beendet, sobald die Variable nicht mehr 1 entspricht.

```
int main() {  
    int zahl = 1;  
    while (zahl == 1) {  
        cout << "Jap, 1 ist immer noch 1." << endl;  
        cout << "Gebe einen neuen Wert ein: ";  
        cin >> zahl;  
    }  
}
```

Fußgesteuerte Schleife

Fußgesteuerte Schleifen werden ein Mal ausgeführt, bevor die Kondition überprüft wird:

```
do {  
    Ausfuehrung  
} while (Kondition)
```

Wenn die Kondition hier immer noch zutrifft, wird die Schleife erneut ausgeführt.

Dieser Code tut im Prinzip das gleiche, wie der Code oben, nur dass die Ausführung immer stattfindet, egal ob die Variable am Anfang den Wert 1 hat oder nicht:

```
do {  
    cout << "Jap, 1 ist immer noch 1!" << endl;  
    cout << "Gib eine neue Zahl ein: ";  
    cin >> zahl;  
} while (zahl == 1);
```

Der gesamte Code sieht so aus:

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int zahl = 1;
6      do {
7          cout << "Jap, 1 ist immer noch 1!" << endl;
8          cout << "Gib eine neue Zahl ein: ";
9          cin >> zahl;
10     } while (zahl == 1);
11     return 0;
12 }
```

Zählergesteuerte Schleife

Eine zählergesteuerte Schleife für einen Prozess eine bestimmte Anzahl an Male aus. Hier wird eine bestimmte Sache 5 Mal ausgeführt:

```
cout << "Nenne eine Zahl, welche 5x plus 1 gerechnet werden soll: ";
cin >> variable;
for (int i = 0; i <= 4; i++) {
    variable++;
    cout << variable << endl;
}
return 0;
```

Selektion

Die Grundlegende Struktur bei einer Selektion sieht so aus:

```
switch (Nummer) {  
    case 1: {  
        Ausfuehrung bei Case 1;  
        break;  
    }  
  
    case 2: {  
        Ausfuehrung bei Case 2;  
        break;  
    }  
  
    case 3: {  
        Ausfuehrung bei Case 3;  
        break;  
    }  
}
```

Und eine komplett funktionierende Situation könnte so aussehen:

```
int Nummer = 0;
cout << "Case 1, 2, oder 3?";
cin >> Nummer;

switch (Nummer) {
    case 1: {
        cout << "Du hast Szenario 1 ausgewaehlt!";
        break;
    }

    case 2: {
        cout << "Du hast Szenario 2 ausgewaehlt!";
        break;
    }

    case 3: {
        cout << "Du hast Szenario 3 ausgewaehlt!";
        break;
    }
}
```

für alles andere wird ein default zweig ausgewählt

```
default: {
    cout << ...;
    break;
}
```

**DENK DARAN DEN CODE MINDESTENS EIN MAL OHNE ABLESEN SELBER ZU SCHREIBEN.
SONST KANNST DU NICHT ÜBERPRÜFEN, OB DU ES WIRKLICH KANNST!!**

Zusatz:

if

```
if (n > 0) {
    cout << "The number is positive";
} else if (n < 0) {
    cout << "The number is negative";
} else {
    cout << "The number is zero.";
}
```