

Referenzen in C++

Neben dem Einsatz von Zeigern bietet C++ eine weitere Möglichkeit, direkt auf den Speicher zuzugreifen, die Arbeit mit Referenzen.

1. Erzeugen einer Referenz

☛ Ergänzen und testen Sie folgendes Programm:

Nach der Deklaration einer Ganzzahl erzeugt man eine Referenz. Dazu wird der Typ gefolgt vom Referenzoperator & (nicht verwechseln mit dem Adressoperator!) und dem Namen der Referenz angegeben. Die Referenz initialisiert man danach sofort mit der Variable zahl. Im Anschluss daran werden Wert und Adresse von zahl und rRef ausgegeben.

```
...  
int zahl = 5;  
int& rRef = zahl;  
  
cout << "zahl: " << zahl << endl;  
cout << "rRef: " << rRef << endl;  
  
cout << "&zahl: " << &zahl << endl;  
cout << "&rRef: " << &rRef << endl;  
...
```

☛ Erklären Sie die Bildschirmausgabe!

Sowohl die Adresse als auch der Wert von zahl und rRef liefern das selbe Ergebnis

☛ Ergänzen Sie das Programm, so dass zahl auf den Wert 7 gesetzt und die Werte bzw. Adressen von zahl und rRef erneut auf dem Bildschirm ausgegeben werden!

C++ bietet keine Möglichkeit, auf die Adresse der Referenz selbst zuzugreifen, da diese im Vergleich zu einem Zeiger oder einer anderen Variablen unerheblich ist. Referenzen werden bei ihrer Deklaration initialisiert und können immer als Synonyme (= andere Bezeichnung für die gleiche Variable) verwendet werden.

2. Regeln zur Anwendung von Referenzen

☛ Fügen Sie dem Programm am Ende eine weitere integer Variable mit der Bezeichnung zahl2 und dem Wert 9 hinzu!

- Weisen Sie der Referenz die Variable zahl2 zu (rRef = zahl2).
- Geben Sie Adresse und Wert von zahl, zahl2 und rRef auf dem Bildschirm aus!

☛ Beschreiben Sie die Ausgabe!

Der Wert von zahl, zahl2, rRef beträgt 9
rRef verweist immer noch auf zahl

Im Programm wird versucht, der Referenz eine neue Variable zuzuweisen. rRef wirkt aber weiterhin als Synonym für zahl. In Wirklichkeit wird also folgende Anweisung ausgeführt:
zahl = zahl2;

Referenzen werden nur einmal initialisiert

Referenzen kontra Zeiger

Wenn man Zeiger löscht oder nicht initialisiert, sollte man ihnen NULL zuweisen, damit kein „wilder Zeiger“ entsteht.

Eine Referenz hingegen **darf nicht** NULL sein! Ein Programm mit einer Referenz auf ein Null – Objekt ist unzulässig. Eine Referenz lässt sich nur einmal initialisieren, d.h. Referenzen auf NULL ergeben in C++ keinen Sinn ☺.

Referenzen dürfen nicht auf NULL gesetzt werden

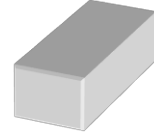
3. Wann benötigt man Referenzen?

Funktionen können in C++ nur **einen** einzigen Wert zurückgeben. Mit Hilfe von Referenzen kann dieses Problem umgangen werden.

Beispiel:

In einem CAD Programm sollen für die Berechnung von Schnittzeichnungen die Flächen der drei unterschiedlichen Seiten eines Quaders berechnet werden. Länge, Breite und Höhe eines Quaders gibt der Benutzer ein.

Die Ergebnisse sollen auf dem Bildschirm ausgegeben werden.



Lösung:

Zur Berechnung der drei Flächen könnte man nun eine Funktion entwerfen und diese in der Hauptfunktion main dreimal mit unterschiedlichen Parametern aufrufen.

Viel Eleganter ist eine Lösung mit Hilfe von Referenzen:

Dazu definiert man die Funktion Quader mit den Parametern rf1 (Fläche1) ... rf3 als Referenzen:

// Berechnet die Flächen eines Quaders:

```
void Quader(double& rf1, double& rf2, double& rf3)
```

Die Eingabe der drei Seiten des Quaders erfolgt in der Funktion Quader(...):

```
    cout << "Laenge? ";
```

```
    cin >> laenge;
```

```
    ...
```

Die Berechnung der drei Flächen nimmt man ebenso in der Funktion Quader(...) vor:

```
    // Berechnung
```

```
    rf1 = laenge * breite;
```

```
    ...
```

Um die Ergebnisse in der Hauptfunktion main() ausgeben zu können benötigt man drei Variablen als Zwischenspeicher:

```
int main( )
```

```
{
```

```
    // Deklaration der Flächen
```

```
    double flaeche1, flaeche2 , flaeche3;
```

```
    ...
```

Um die Berechnung auszuführen, übergibt man der Funktion Quader die Parameter für flaeche1 bis flaeche3 als Referenzen:

```
    // Berechnung ausführen
```

```
    Quader(flaeche1, flaeche2, flaeche3);
```

Zuletzt gibt man die Ergebnisse auf dem Bildschirm aus:

```
    // Ergebnisse ausgeben
```

```
    cout << "Flaeche 1: " << flaeche1 << endl;
```

```
    ...
```

☛ **Ergänzen Sie das Programm, und testen Sie seine Funktion!**

☛ **Erklären Sie die Funktionsweise des Programms!**

☛ **Ändern Sie das Programm, so dass die Flächenberechnung solange ausgeführt wird bis der Benutzer <e>nde eingibt!**