

Pointer in C(++)

Geschrieben von:

Sonntag, den 18. Dezember 2005 um 18:02 Uhr - Aktualisiert Mittwoch, den 10. Oktober 2007 um 21:03 Uhr

Dieser Artikel wird sich mit Zeigern (engl. Pointer) in C, bzw. C++ beschäftigen. Der Quellcode wurde mit dem g++ 3.3.6 getestet. Sollten sich trotzdem irgendwelche groben Fehler eingeschlichen haben, bitte ich um Mitteilung.

Folgendes wird behandelt:

1. Einführung
2. Zugriff auf Daten mit Pointer
3. Übergabe/Rückgabe von Pointer
4. Referenzen
5. Zeigerarithmetik
6. const bei Pointer
7. Speicher allokieren
8. Pointer auf Pointer

1 Einführung

Viele verteufeln sie als fehleranfällig und kompliziert, aber wir brauchen sie trotzdem: Pointer, z.B. um Datenstrukturen wie Listen oder Bäume effizient zu implementieren oder einfach um das Kopieren eines Objekts bei einer Übergabe zu vermeiden.

Für Einsteiger ist das Durcheinander der Sternchen und &-Zeichen meistens recht verwirrend. Mal sehen, ob wir etwas Klarheit reinbringen können...

Zuerst werden wir uns anhand eines kurzen Beispiels anschauen, was normale Variablen und Pointervariablen im Speicherbild unterscheidet:

```
[code xml:lang="cpp"]#include <string> using namespace std; int main(int argc, char **argv) {  
int i = 5; // int Variable int *p = 0; // Pointer auf int, mit 0 initialisiert! p = &i; // mit Operator  
& Adresse von i holen und an p zuweisen cout
```