

Zeiger auf Funktionen

Der Name einer Funktion bezeichnet ihre eigene Startadresse im Arbeitsspeicher.

Der Funktionsaufruf einer Funktion mittels einer Zeigervariablen entspricht dem eines gewöhnlichen Funktionsaufrufes.

Wozu?

Zeiger auf Funktionen bieten mehr Möglichkeiten, Funktionen aufzurufen.

z.B. Zeiger auf Funktionen ermöglichen einem Programm, zwischen mehreren Funktionen auszuwählen und sich für die Funktion zu entscheiden, die unter den gegebenen Umständen am geeignetsten ist.

Zeiger auf Funktionen deklarieren

Die allgemeine Syntax:

```
datentyp (*zgr_auf_funk) (parameter_liste);
```

Obige Anweisung deklariert **zgr_auf_funk** als Zeiger auf eine Funktion die den Rückgabetyt **typ** hat und der die Parameter aus **parameter_liste** übergeben werden.

Konkrete Beispiele zur Deklaration:

```
int (*funk1) (int x); // deklariert funk1 als Zeiger auf eine Funktion, die ein Argument vom  
// Typ int übernimmt und einen Wert vom Typ int zurückliefert.
```

```
void (*funk2) (double y, double z); // deklariert funk2 als Zeiger auf eine Funktion, die zwei  
// double-Argumente übernimmt und void als Rückgabetyt hat  
// (also keinen Wert zurückliefert)
```

```
char (*funk3) (char *p[]); // deklariert funk3 als Zeiger auf eine Funktion, die ein Array  
// von Zeigern auf char als Argument übernimmt und einen  
// Wert vom Typ char zurückliefert
```

```
void (*funk4) (); // deklariert funk4 als Zeiger auf eine Funktion, die kein  
// Argument übernimmt und void als Rückgabetyt hat
```

Wichtig: Wenn Sie einen Zeiger auf eine Funktion deklarieren, muss der Zeigernamen und der Indirektionsoperator ***** in Klammern gesetzt werden.

```
int *funk1(int x); // Fehler bei der Deklaration  
// Obige Deklaration ohne Klammern um den Zeigernamen deklariert  
// daher funk1 als eine Funktion, die einen Zeiger auf int zurückliefert
```

Der Grund für die Klammern ist die relativ niedrige Priorität des Indirektionsoperators *****. Die Priorität des Indirektionsoperators liegt noch unter der Priorität der Klammern für die Parameterliste.

Zeiger auf Funktionen initialisieren und verwenden

Neben der Deklaration des „Zeigers auf eine Funktion“, muss der Zeiger initialisiert werden, damit er auf eine Funktionverweist.

Wichtig ist, dass Rückgabetyt und Parameterliste der Funktion mit dem Rückgabetyt und der Parameterliste aus der Zeigerdeklaration übereinstimmen.

Beispiel: Der folgende Code deklariert und definiert eine Funktion und einen Zeiger auf diese Funktion:

```
float quadrat(float x);    // Der Funktionsprototyp
float (*p) (float x);    // Die Zeigerdeklaration

float quadrat(float x)    // Die Funktionsdefinition
{
    return x * x;
}
:
p = quadrat;    // Da die Funktion quadrat() und der Zeiger p die gleichen Parameter und
                // Rückgabetypen haben, können Sie p so initialisieren, dass er auf quadrat zeigt
:
antwort = p(x);    // Aufruf der Funktion über den Zeiger
```

// Beispiel für Deklaration und Einsatz eines Funktionszeigers
// Die Funktion wird zweimal aufgerufen, einmal über den Funktionsnamen und beim zweiten Mal
// über den Zeiger. Beide Aufrufe führen zu demselben Ergebnis.

```
#include <stdio.h>
double quadrat(double x);    // Der Funktionsprototyp
double (*p)(double x);    // Die Zeigerdeklaration

int main(void)
{
    p = quadrat;    // p wird mit quadrat initialisiert
    printf("%f %fn", quadrat(6.6), p(6.6));    // quadrat() wird auf zwei Wegen aufgerufen
    return(0);
}

double quadrat(double x)
{
    return x * x;
}
```

Ausgabe: 43.560000 43.560000