

Funktionszeiger

In C können Funktionen über Zeiger aufgerufen werden, die die Aufruf-Adresse der Funktion enthalten.

Ein Funktionszeiger wird folgendermaßen deklariert:

*typ (*f)();*

f ist der Name der Funktion und *typ* der Typ des Rückkehrwertes.
Der Funktionszeiger muss aber unbedingt mit der Adresse einer definierten Funktion belegt werden!

Funktionszeiger

```
void main()
{
    int i,dim; float mittel,messwerte[100];
    float (*fk)(); //Deklaration eines Funktionszeigers fk

    fk=&amittel; //Zuweisen der Aufrufadresse der definierten Funktion amittel

    printf("\nAnzahl Werte=");
    scanf("%d",&dim);

    for (i=0;i<dim;i++)
    { printf("\nwert[%d]=",i);
      scanf("%f",&messwerte[i]);
    }

    mittel=(*fk)(dim, messwerte) ; //Aufruf der Funktion amittel
    printf("\narithm.Mittel=%f",mittel);
}
```

R.Großmann, P. Sobe

Funktionen als Parameter von Funktionen

In C müssen Funktionen, die als Parameter an andere Funktionen übergeben werden sollen, in der Parameterliste als **Funktionszeiger** gekennzeichnet werden. Zusätzlich zur normalen Funktionszeiger-notation müssen auch die Typen der Parameter dieser zu übergebenden Funktion aufgeführt sein.

Beispiel: *float (*f)(float)*

das wäre der Funktionszeiger einer Funktion *f(x)*, die einen **float-Parameter** x hat und einen float-Wert zurückgibt (z.B. **cos(x)**).

```
void meine_funktion( ... , float (*f)(float) , ... ) { ... }
```

Anwendungsbereiche:

Funktionen müssen an Funktionen übergeben werden bei Berechnung von beispielsweise

- Nullstellenberechnung von Funktionen
- bestimmten Integralen
- Interpolation von Funktionen

Beispiel für Funktionen als Parameter von Funktionen

//Halbierungsverfahren zur Nullstellenberechnung

```
float halb(float a, float b, float (*f)(float) )  
{ float m;  
  m=(a+b)/2.0f;  
  while (fabs( (*f)(m) )>0.00001)  
  { m=(a+b)/2.0f;  
    if ( (*f)(m) * (*f)(b) < 0.0f)  
      a=m;  
    else  
      b=m;  
  }  
  return m;  
}
```

Beispiel für Funktionen als Parameter von Funktionen

```
float funkt(float x)
{ return cos(x); }
```

```
void main()
{ float a,b,nullst;
  printf("Halbierungsverfahren\n");
  printf("Eingabe Intervall\n a=");
  scanf("%f",&a);
  printf(" b=");
  scanf("%f",&b);
  nullst=halb(a,b, funkt); //Übergabe der Funktion funkt
  printf("\nNullstelle=%f\n",nullst);
}
```