

## 8 Zeiger auf Funktionen

Mit Hilfe von Zeigern können auch Funktionen angesprochen werden. Damit ist es unter anderem möglich, Funktionen als Parameter zu übergeben oder zur Laufzeit festzulegen.

**typ1 (\*pointer) (typ2);** vereinbart einen Zeiger *pointer* auf eine Funktion mit Rückgabewert vom Typ *typ1*. Das erste Klammerpaar ist nötig, um *pointer* von einem Funktionsnamen abzugrenzen.

**pointer = funct;** ein Zeiger *pointer* auf eine Funktion wird mit der Funktion *funct* initialisiert.

**pointer(par);** Aufruf einer Funktion mit dem Parameter *par* über einen Pointer. Das kann auch mit **(\*pointer)(par);** erfolgen.

Listing 40: Pointer auf Funktion (funcpointer.c)

```
#include <stdio.h>

int square(int x)
{
    return(x*x);
}

int main()
{
    int (*psquare) (int), i;
    psquare = square;
    for ( i = 0; i <= 10; i++ ) {
        printf("%i^2 = %i\n", i, psquare(i));
    }
    return 0;
}
```

Listing 41: Pointer auf Funktion (multfuncts.c)

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    double x = 0.0;
    double (*pfunc) (double);
    /* double sin(double), cos(double), tan(double); */
    int i;
    char c;
    printf("Waehlen Sie fuer sinus\t\t s\n");
    printf("Waehlen Sie fuer cosinus\t c\n");
    printf("Waehlen Sie fuer tans\t\t t\n");
    (void) scanf("%c", &c);
    switch(c)
    {
        case 's' : pfunc = sin;
                    break;
        case 'c' : pfunc = cos;
                    break;
        case 't' : pfunc = tan;
                    break;
        default: printf("Falsche Eingabe\n");
    }
    for ( i = 0; i <= 10; i++){
        x += .3;
        printf("%c(%f) = %f\n", c, x, pfunc(x));
    }
    return 0;
}
```

Listing 42: Pointer auf Funktion (funcpointer1.c)

```
#include <stdio.h>

static int square(int), cube(int);
static void compute(int, int (int), int (int));

int square(int x)
{
    return (x * x);
}

int cube(int x)
{
    return (x * x * x);
}

void compute(int n, \
             int (*square) (int), int (*cube) (int))
{
    int i;
    for (i = 0; i <= n; i++)
        printf("%5i | %5i | %5i \n", i, square(i), cube(i));
}

int main(void)
{
    int n;
    printf("Eingabe Zahl der Auswertungen: ");
    (void) scanf("%i", &n);
    compute(n, square, cube);
    return 0;
}
```