

Praktikum Programmieren, AI-SS09

Übung-Nr. 4

Aufgabe: *Ein einfacher Taschenrechner*

Bearbeiter: *T. Merkel* Matr. Nr. 19869

F. Rosenkranz Matr. Nr. 19895

Abgegeben am 28.04.09

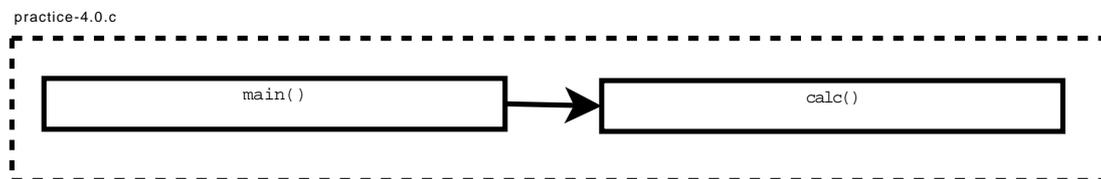
-
- Aufgabe erfüllt
 - Bitte korrigieren:
 - Einrückung des Quellcodes
 - Sprechende Namen(Bezeichner) für Variable und Funktionen
 - Nur eine Variable pro Zeile definieren
 - Nur eine Anweisung pro Zeile
 - Ablaufdiagramm
 - Testfälle
 - weitere Anmerkungen
-

1 Beschreibung der Lösung

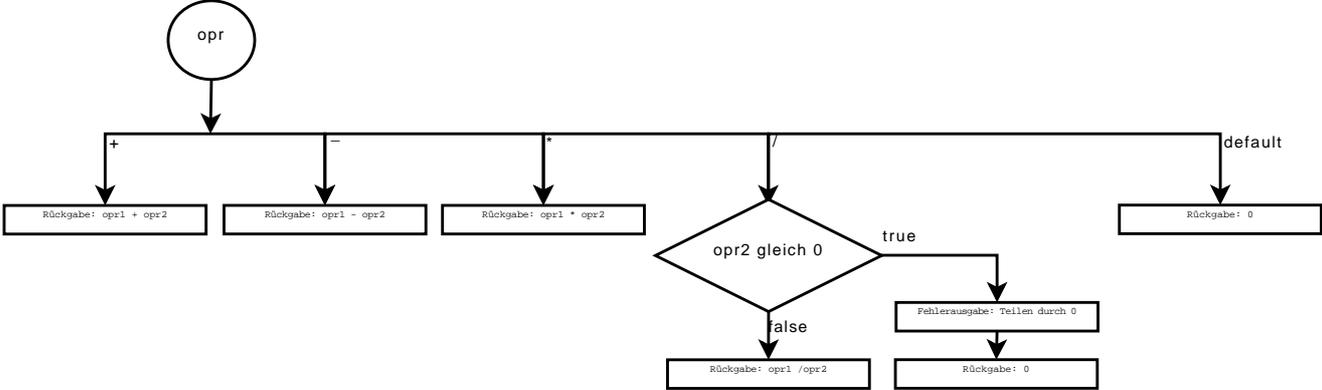
Der Taschenrechner liest per Benutzereingabe den Operator und zwei Operanden ein. Zur Berechnung wird die *calc()*-Funktion aufgerufen, diese wählt dann per *switch*-Anweisung den entsprechenden Rechenschritt aus. Das Ergebnis wird an die *main()*-Funktion zurück gegeben. Diese zeigt das Ergebnis dann am Bildschirm an.

2 Diagramme

2.1 Aufrufhierarchie



2.2 Berechnungsfunktion



3 Quellcode

```
1  /*-----
2  * $Id: practice-4.0.c,v 0.6 2009/04/22 14:05:38 tmerkel Exp $
3  * Copyright 2009 HS-Weingarten (thomas.merkel@hs-weingarten.de)
4  *-----
5  */
6
7  #include <stdio.h>
8
9  /* function:    calc()
10 * description:  calculate two operands
11 * parameters:   optr - operator
12 *              opr1 - operand 1
13 *              opr2 - operand 2
14 * returns:      float value of the calc result
15 *
16 */
17 float calc( char optr, float opr1, float opr2 ) {
18     /* switch to the right operator */
19     switch( optr ) {
20         case '+': return opr1 + opr2;
21         case '-': return opr1 - opr2;
22         case '*': return opr1 * opr2;
23         case '/':
24             /* check for nullity */
25             if( opr2 == 0 ) {
26                 printf( "Fehler, division durch 0 nicht erlaubt\n" );
27                 return 0;
28             } else {
29                 return opr1 / opr2;
30             }
31         default: return 0;
32     }
33 }
34
35 int main( void ) {
36     float opr1, opr2, ret;
37     char optr;
38     while( 1 ) {
39         /* user input */
40         fpurge( stdin );
41         printf( "Eingabe: Operator <return> Operand1 <return> Operand2 <return>\n" );
42         printf( "Bitte eingabe Operator (+, -, *, /, q): " );
43         scanf( " %c", &optr );
44
45         /* check user input, if 'q' quit the program */
46         if( optr == 'q' )
47             return 0;
48
49         /* check user input for missing operand */
```

```

50     if(( optr != '+' ) && ( optr != '-' ) && ( optr != '*' ) && ( optr != '/' )) {
51         printf( "Fehlerhafter Operator!\n" );
52         continue;
53     }
54
55     printf( "Bitte Eingabe Operand1: " );
56     scanf( "%f", &opr1 );
57     printf( "Bitte Eingabe Operand2: " );
58     scanf( "%f", &opr2 );
59
60     /* call calc function */
61     ret = calc( optr, opr1, opr2 );
62
63     /* output the calc result */
64     printf( "Ergebnis: %.2f %c %.2f = %.2f\n", opr1, optr, opr2, ret );
65 }
66 }

```

4 Beispielausgabe

```

Bitte eingabe Operator (+, -, *, /, q): *
Bitte Eingabe Operand1: 13
Bitte Eingabe Operand2: 13
Ergebnis: 13.00 * 13.00 = 169.00

```

5 Testfälle

Eingabe Operator und Operanden	Ausgabe Ergebnis
+ 3 5	8.00
- b	erneute Benutzereingabe
/ 5 0	erneute Benutzereingabe