

Lösung Übungszettel 6 Aufgabe 1-4

1 Aufgabe 1 und 2

1.1 Die Bibliothek *biblio.c*

```
/* Bibliothek zum Addieren und Subtrahieren von  
Integer- und Double-Zahlen  
*/
```

```
//Funktion zum Addieren von Integer-Zahlen  
//1. Zahl a  
//2. Zahl b  
//Rückgabewert a+b  
int addInt(int a, int b)  
{  
    return a+b;  
}
```

```
//Funktion zum Subtrahieren von Integer-Zahlen  
//1. Zahl a  
//2. Zahl b  
//Rückgabewert a-b  
int subInt(int a, int b)  
{  
    return a-b;  
}
```

```
//Funktion zum Addieren von Double-Zahlen  
//1. Zahl a  
//2. Zahl b  
//Rückgabewert a+b  
double addDouble(double a, double b)  
{  
    return a+b;  
}
```

```
//Funktion zum Subtrahieren von Double-Zahlen  
//1. Zahl a  
//2. Zahl b  
//Rückgabewert a-b  
double subDouble(double a, double b)  
{
```

```
    return a-b;
}
```

1.2 Die Header-Datei *biblio.h*

```
/* Header-Datei zur Bibliothek biblio.c */

//Funktionsprototypen definieren
double addDouble(double a, double b);
double subDouble(double a, double b);
int addInt(int a, int b);
int subInt(int a, int b);
```

1.3 Das Programm *addieren.c*

```
/* Anwendung der Bibliothek biblio.c */

//<stdio.h> einbinden für Ausgabe mit printf
//"biblio.h" einbinden für die Nutzung von biblio.c
#include <stdio.h>
#include "biblio.h"

int main()
{
    //Double- und Integer-Zahlen deklarieren und initialisieren
    double a = 5.3, b = 3.11;
    int c = 1, d = 5;

    //Die Funktionen der Bibliothek benutzen und Ergebnisse ausgeben
    printf("%g + %g = %g\n", a, b, addDouble(a,b));
    printf("%g - %g = %g\n", a, b, subDouble(a,b));
    printf("%d + %d = %d\n", c, d, addInt(c,d));
    printf("%d - %d = %d\n", c, d, subInt(c,d));
}
```

2 Aufgabe 3

2.1 Die Bibliothek *biblio2.c*

```
/* Bibliothek zum Addieren und Subtrahieren von
   Integer- und Double-Zahlen
   Allgemeine Version mit void-Zeigern
*/

//Deklariere von globalen Variablen für die Ergebnisse
//lokale Variablen werden gelöscht, wenn eine Funktion
//abgearbeitet ist, deshalb zeigt ein Zeiger auf eine
//lokale Variable dann ins Nichts
double ergebnis1;
int ergebnis2;

//Funktion zum Addieren
//typ zum Feststellen des Types
//('i' = Integer, 'f' = Double)
//a ist Zeiger auf erste Zahl
//b ist Zeiger auf zweite Zahl
//Rückgabewert: Zeiger auf Ergebnis
void *add(char typ, void *a, void *b)
{
    switch(typ)
    {
        case 'i':
            //1. Fall: Integer
            //Ergebnis berechnen, Casting für Integer aus void!
            ergebnis2 = *(int *)a + *(int *)b;
            //Ergebnis zurückgeben
            return &ergebnis2;
        case 'f':
            //2. Fall: Double
            //Ergebnis berechnen, Casting für Double aus void!
            ergebnis1 = *(double *)a + *(double *)b;
            //Ergebnis zurückgeben
            return &ergebnis1;
    }
}

//Funktion zum Subtrahieren
//typ zum Feststellen des Types
//('i' = Integer, 'f' = Double)
//a ist Zeiger auf erste Zahl
//b ist Zeiger auf zweite Zahl
//Rückgabewert: Zeiger auf Ergebnis
void *sub(char typ, void *a, void *b)
{
    switch(typ)
    {
```

```

    case 'i':
        //1. Fall: Integer
        //Ergebnis berechnen, Casting für Integer aus void!
        ergebnis2 = *(int *)a - *(int *)b;
        return &ergebnis2;
    case 'f':
        //2. Fall: Double
        //Ergebnis berechnen, Casting für Double aus void!
        ergebnis1 = *(double *)a - *(double *)b;
        return &ergebnis1;
    }
}

```

2.2 Die Header-Datei *biblio2.h*

```

/* Header-Datei zur Bibliothek biblio2.c */

//Funktionsprototypen definieren
void *add(char typ, void *a, void *b);
void *sub(char typ, void *a, void *b);

```

2.3 Das Programm *addieren2.c*

```

/* Anwendung der Bibliothek biblio2.c */

//<stdio.h> einbinden fuer Ausgabe mit printf
//"biblio2.h" einbinden fuer die Nutzung von biblio2.c
#include <stdio.h>
#include "biblio2.h"

int main()
{
    //Double- und Integer-Zahlen deklarieren und initialisieren
    double a = 74.3, b = 3.11;
    int c = 12, d = 5;

    //Die Funktionen der Bibliothek benutzen und Ergebnisse ausgeben
    //Achtung: Casting der Ergebnisse für die Ausgabe!
    printf("%g + %g = %g\n", a, b, *(double *)add('f', &a, &b));
    printf("%g - %g = %g\n", a, b, *(double *)sub('f', &a, &b));
    printf("%d + %d = %d\n", c, d, *(int *)add('i', &c, &d));
    printf("%d - %d = %d\n", c, d, *(int *)sub('i', &c, &d));
}

```

3 Aufgabe 4

3.1 Die Bibliothek *biblio3.c*

```
/* Bibliothek zum Addieren und Subtrahieren von
   Integer- und Double-Zahlen und Zeichenketten
   Allgemeine Version mit void-Zeigern
*/

//Einbinden von <string.h> fuer die Bearbeitung von Zeichenketten
#include <string.h>

//Deklariere von globalen Variablen für die Ergebnisse
//lokale Variablen werden gelöscht, wenn eine Funktion
//abgearbeitet ist, deshalb zeigt ein Zeiger auf eine
//lokale Variable dann ins Nichts
double ergebnis1;
int ergebnis2;
char *ergebnis3;

//Funktion zum Addieren
//typ zum Feststellen des Types
//('i' = Integer, 'f' = Double, 's' = String)
//a ist Zeiger auf ersten Summand
//b ist Zeiger auf zweiten Summand
//Rückgabewert: Zeiger auf Ergebnis
void *add(char typ, void *a, void *b)
{
    //Variable fuer die Zeichenlaenge der Teilstrings
    int zeichen;

    switch(typ)
    {
        case 'i':
            //1. Fall: Integer
            //Ergebnis berechnen, Casting für Integer aus void!
            ergebnis2 = *(int *)a + *(int *)b;
            //Ergebnis zurueckgeben
            return &ergebnis2;
        case 'f':
            //2. Fall: Double
            //Ergebnis berechnen, Casting für Double aus void!
            ergebnis1 = *(double *)a + *(double *)b;
            //Ergebnis zurueckgeben
            return &ergebnis1;
        case 's':
            //3. Fall: String
            //Laenge des Ergebnisstrings berechnen:
            //Laenge des ersten Teilstrings +
            //Laenge des zweiten Teilstrings + 1 für das Endzeichen \0
            zeichen = strlen((char *)a) + strlen((char *)b);
```

```

        //Speicher fuer den Ergebnisstring bereitstellen
        ergebnis3 = (char *)malloc(zeichen * sizeof(char) -1);
        //Ersten Teilstring in den Ergebnisstring kopieren
        strcpy(ergebnis3, (char *) a);
        //Zweiten Teilstring anhaengen
        strcat(ergebnis3, (char *) b);
        //Ergebnis zurueckgeben
        return ergebnis3;
    }
}

//Funktion zum Subtrahieren
//typ zum Feststellen des Types
//(‘i’ = Integer, ‘f’ = Double)
//a ist Zeiger auf erste Zahl
//b ist Zeiger auf zweite Zahl
//Rückgabewert: Zeiger auf Ergebnis
void *sub(char typ, void *a, void *b)
{
    switch(typ)
    {
        case ‘i’:
            //1. Fall: Integer
            //Ergebnis berechnen, Casting für Integer aus void!
            ergebnis2 = *(int *)a - *(int *)b;
            //Ergebnis zurueckgeben
            return &ergebnis2;
        case ‘f’:
            //2. Fall: Double
            //Ergebnis berechnen, Casting für Double aus void!
            ergebnis1 = *(double *)a - *(double *)b;
            //Ergebnis zurueckgeben
            return &ergebnis1;
    }
}

```

3.2 Die Header-Datei *biblio3.h*

```

/* Header-Datei zur Bibliothek biblio3.c */

//Funktionsprototypen definieren
void *add(char typ, void *a, void *b);
void *sub(char typ, void *a, void *b);

```

3.3 Das Programm *addieren3.c*

```

/* Anwendung der Bibliothek biblio3.c */

//<stdio.h> einbinden für Ausgabe mit printf
//"biblio3.h" einbinden für die Nutzung von biblio3.c

```

```
#include <stdio.h>
#include "biblio3.h"

int main()
{
    //Double- und Integer-Zahlen deklarieren und initialisieren
    double a = 73.3, b = 3.11;
    int c = 121, d = 8;
    //Zwei Strings zum Addieren deklarieren und initialisieren
    char string1[7] = "Hello ";
    char string2[6] = "World";

    //Die Funktionen der Bibliothek benutzen und Ergebnisse ausgeben
    printf("%g + %g = %g\n", a, b, *(double *)add('f', &a, &b));
    printf("%g - %g = %g\n", a, b, *(double *)sub('f', &a, &b));
    printf("%d + %d = %d\n", c, d, *(int *)add('i', &c, &d));
    printf("%d - %d = %d\n", c, d, *(int *)sub('i', &c, &d));
    printf("%s + %s = %s\n", string1, string2, (char *)add('s', string1, string2));
}
```