

ALGORITHMEN UND DATENSTRUKTUREN

ÜBUNG 5: FUNKTIONEN & PULSIERENDER SPEICHER

Eric Kunze
eric.kunze@tu-dresden.de

TU Dresden, 18. November 2021

PULSIERENDER SPEICHER

Gültigkeitsbereiche von Objekten:

- Eine Funktion ist ab ihrer Deklaration bis zum
 Programmende sichtbar. Vorwärtsdeklarationen beachten!
- Ihre formalen Parameter jedoch nur innerhalb der Funktionsdefinition!
- Gibt es gleichlautende formale Parameter in verschiedenen Funktionen, müssen diese in der Tabelle natürlich unterschieden werden (z.B. durch "x in f").
- Vorsicht bei Namenskonflikten: lokale Variablen überschreiben die Sichtbarkeit globaler Variablen.

PULSIERENDER SPEICHER

Speicherprotokoll:

- Für jeden Funktionsaufruf werden erst die Parameter, dann die lokalen Variablen in Reihenfolge ihres Auftretens in der Umgebung notiert. Globale Variablen stehen ganz vorn.
- Variablennamen werden nur notiert, wenn die Variablen sichtbar sind. Globale Variablennamen werden immer notiert.
- Der Wert von nicht sichtbaren Variablen muss nur notiert werden wenn er sich ändert.
- Uninitialisierte Variablen werden mit Inhalt "?" notiert.

AUFGABE 3

```
1 #include <stdio.h>
                                              while (a != 1) {
                                         18
   int a:
                                         19
                                               f(&i, a):
                                                                   /*$2*/
3
                                         20
                                               a = a / 2:
                                               *b = *b + 1:
   void g(int a, int *b);
                                         21
5
                                         22
                                            /*label4*/
   void f(int *i, int j) {
                                         23
   /*label1*/
                                            }
7
                                         24
    if (*i + i < a) {
                                         25
9
    *i = *i + 1;
                                         26
                                            int main() {
                                              int x = 0:
10
    f(i, j);
                         /*$1*/
                                         27
                                             scanf("%i", &a):
11
                                         28
12
     /*label2*/
                                         29
                                             /*label5*/
13
                                         g(a, \&x);
                                                                   /*$3*/
14
                                         31 /*label6*/
   void g(int a, int *b) {
                                         32
                                             return 0;
15
16
     int i = 2;
                                         33 }
    /*label3*/
17
```

AUFGABE 3 — TEIL (A)

Gültigkeitsbereiche

Objektname	Gültigkeitsbereich
a	2 – 14 und 25 – 33
g	4 - 33
a, bing	15 – 24
i in g	16 – 24
f	6 – 33
i,j in f	6 – 13
main	26 - 33
x in main	27 – 33

AUFGABE 3 — TEIL (B)

Label	RM	1	2	3	4	5	6	7	8	9	10	11	12

AUFGABE 3 — TEIL (B)

Label	RM	1	2	3	4	5	6	7	8	9	10	11	12
label5	-	a 7	х 0										
label3	3			a 7	b 2	i 2							
label1	2:3	a 7					i 5	j 7					
label2	2:3	a 7					i 5	j 7					
label4	3		1	a 3	b 2	i 2							
label1	2:3	a 7					i 5	ј 3					
label1	1:2:3	a 7				3			i 5	ј 3			
label1	1:1:2:3	a 7				4					i 5	ј 3	
label2	1:1:2:3	a 7									i 5	ј 3	
label2	1:2:3	a 7							i 5	ј 3			
label2	2:3	a 7					i 5	ј 3					
label4	3		2	a 1	b 2	i 4							
label6	_	a 7	х 2										