

## Übung 10

Gegeben ist die nachfolgende Java-Methode als symbolischer Byte-Code:

```
1 public int func(int a, int b);  
2  
3 0:  iload_1  
4 1:  iload_2  
5 2:  imul  
6 3:  iconst_2  
7 4:  imul  
8 5:  istore_3  
9 6:  iload_1  
10 7:  iload_2  
11 8:  isub  
12 9:  istore 4  
13 10: iload_3  
14 11: iload 4  
15 12: iload 4  
16 13: imul  
17 14: isub  
18 15: ireturn
```

### Aufgabe 1

Geben Sie den zugehörigen Java-Quellcode an.

**Lösung:**

```
1 public int func(int a, int b) {  
2     int c = 2 * a * b;  
3     int d = a - b;  
4     return c - d * d;  
5 }
```

## Aufgabe 2

Optimieren Sie die Methode, so dass Sie mit möglichst wenig Befehlen und Speicherzugriffen auskommen.

### Lösung:

```
1 public int func(int a, int b);
2
3 0: iload_1
4 1: iload_2
5 2: imul
6 3: iconst_2
7 4: imul
8 5: iload_1
9 6: iload_2
10 7: isub
11 8: dup
12 9: imul
13 10: isub
14 11: ireturn
```

Gegeben ist eine Funktion mit symbolischen Maschinenbefehlen für die Java-VM:

```
1 public static int fact(int);
2 Code:
3 0: iload_0
4 1: ifne 4
5 2: iconst_1
6 3: ireturn
7 4: iload_0
8 5: iload_0
9 6: iconst_1
10 7: isub
11 8: invokestatic #2; // Method fact:(I)I
12 9: imul
13 10: ireturn
```

## Aufgabe 3

Stellen Sie den Ablauf und den Rechenstack für  $x = 3$  dar.

### Lösung:

Hinweise: Bei statischen Methoden wird keine Referenz auf das Objekt benötigt, daher existiert auch kein this-Pointer auf dem Stack einer statischen Methode. Jede Spalte stellt den Stack nach Ausführung der Quellcode-Zeile dar, deren Nummer über der entsprechenden Spalte steht. *S* stellt den Zustand vor Ausführung des ersten Befehls des letzten Funktionsaufrufes dar. Die erste Spalte (SF) nennt den aktuellen Stack Frame, dahinter bezeichnen LV und OP die Bereiche für lokale Variablen und den Operanden-Stack.

			S	0	1	4	5							
SF 1	OP	2												
		1												2
		0		2		2	2					2		
	LV	0	x = 2	x = 2	x = 2	x = 2	x = 2							

			S	0	1	4	5	6	7					
SF 0	OP	2						1						
		1					3	3	2					
		0		3		3	3	3	3	3	3	3	3	
	LV	0	x = 3	x = 3	x = 3	x = 3	x = 3	x = 3	x = 3	x = 3	x = 3	x = 3	x = 3	

Abbildung 1: Lösung Aufgabe 3 Teil 1

## Aufgabe 4

Geben Sie die Methode im Java Quellcode an.

**Lösung:**

```

1 public static int fact(int x) {
2     if (x == 0) {
3         return 1;
4     }
5     return x*fact(x-1);
6 }

```

Fortsetzung

Fortsetzung

			S01											
SF 3	OP	2												
		1												
		0	0											
	LV	0	x = 0	x = 0	x = 0									
			S014567											
SF 2	OP	2												
		1	110											
		0	1111111											
	LV	0	x = 1	x = 1	x = 1	x = 1	x = 1	x = 1	x = 1	x = 1	x = 1	x = 1	x = 1	
			67											
SF 1	OP	2	1											
		1	2	1										
		0	2	2	2	2	2	2	2	2	2	2	2	
	LV	0	x = 2	x = 2	x = 2	x = 2	x = 2	x = 2	x = 2	x = 2	x = 2	x = 2	x = 2	
SF 0	OP	2												
		1												
		0	3	3	3	3	3	3	3	3	3	3	3	
	LV	0	x = 3	x = 3	x = 3	x = 3	x = 3	x = 3	x = 3	x = 3	x = 3	x = 3	x = 3	

Abbildung 2: Lösung Aufgabe 3 Teil 2

Fortsetzung			2	3		
SF 3	OP	2				
		1				
		0	1		= 1	
	LV	0	x = 0			
				8	9	10
SF 2	OP	2				
		1		1		
		0	1	1	1	1
	LV	0	x = 1	x = 1	x = 1	x = 1
				8	9	10
SF 1	OP	2				
		1			1	
		0	2	2	2	2
	LV	0	x = 2	x = 2	x = 2	x = 2
				8	9	10
SF 0	OP	2				
		1			2	
		0	3	3	3	3
	LV	0	x = 3	x = 3	x = 3	x = 3

Abbildung 3: Lösung Aufgabe 3 Teil 3