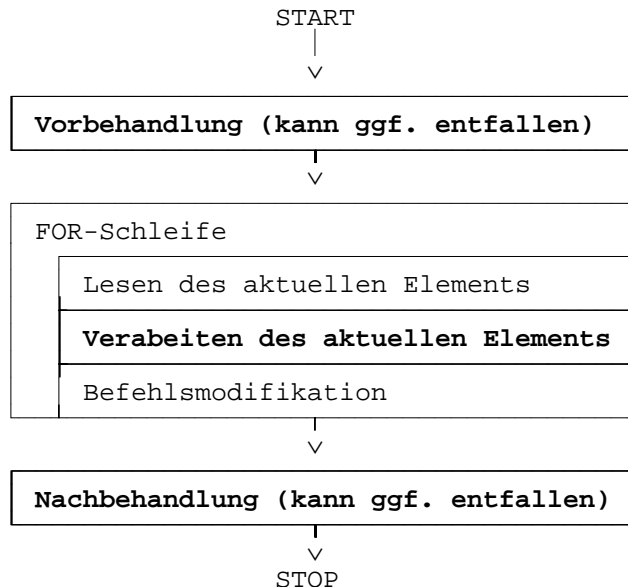


# LC1-Programmierung - "Kochrezept" für eine Problemklasse

## 1. Zielstellung

"Kochrezept" für die vereinheitlichte Behandlung der Problemklasse **Elementweises Lesen und Verarbeiten eines Arrays**. Ausgangspunkt ist eine Art PAP, der die Problemklasse abdeckt:



Vor- und Nachbehandlung sind einmalige Manipulationen am Anfang und am Ende des Programms. Problemabhängig entfallen ggf. Vor- und/oder Nachbehandlung.

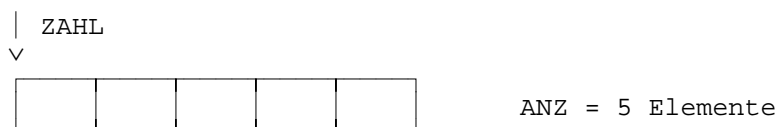
**Beispiel 1:** Es sei die Anzahl der Elemente  $x_i$  eines Arrays zu ermitteln, für die gilt  $|x_i| = |y|$ . Die Betragsbildung für  $y$  ist nur einmal, vor Beginn der Arrayverarbeitung nötig.

**Beispiel 2:** Das Ergebnis einer Berechnung sei in A abzulegen. Da A im Programm belegt ist, wird man das Ergebnis zunächst in einer Speicherzelle berechnen und deren Inhalt erst am Ende nach A transportieren.

Problemabhängig sind nur die "Versatzstücke" Vorbehandlung, Verarbeiten des aktuellen Elements und Nachbehandlung. Der Rest ist problemunabhängig.

## 2. Beispiel

Die Inhalte der Felder eines Arrays seien zu summieren und die Summe auf Adresse SUMME abzulegen. Das Array beginne an der Adresse ZAHL und umfasse (hier) ANZ = 5 Elemente (vgl. Übung 5).



Vor- und Nachbehandlung entfallen; die Nachbehandlung nur dann, wenn man die Summe bereits in der FOR-Schleife auf Adresse SUMME auflaufen lässt.

**Schritt 1:** Vereinbarung der Daten (Konstanten, Variablen), am Ende des Programms

<b>EINS:</b>	<b>DEF</b>	<b>1</b>	<— ist fast immer nötig
<b>;</b>			
<b>SUMME:</b>	<b>DEF</b>	<b>0</b>	
<b>;</b>			
<b>ANZ:</b>	<b>DEF</b>	<b>5</b>	
<b>ZAHL:</b>	<b>DEF</b>	<b>4</b>	
	<b>DEF</b>	<b>-8</b>	
	<b>DEF</b>	<b>11</b>	
	<b>DEF</b>	<b>39</b>	
	<b>DEF</b>	<b>54</b>	

**Schritt 2:** Programmieren der FOR-Schleife als Programmrahmen

<b>M1:</b>	<b>LDA</b>	<b>ANZ</b>	<b>;</b>	<b>A := &lt;ANZ&gt;</b>
	<b>LDB</b>	<b>EINS</b>	<b>;</b>	<b>B := 1</b>
	<b>SUB</b>		<b>;</b>	<b>A := A - B</b>
	<b>MOV</b>	<b>ANZ</b>	<b>;</b>	<b>&lt;ANZ&gt; := A</b>
	<b>JPS</b>	<b>M99</b>	<b>;</b>	<b>IF A &lt; 0 GOTO M99</b>
<b>;</b> Lesen des aktuellen Elements				
<b>;</b> Verarbeiten des aktuellen Elements				
<b>;</b> Befehlsmodifikation				
	<b>JMP</b>	<b>M1</b>	<b>;</b>	<b>GOTO M1</b>
<b>M99:</b>	<b>HLT</b>		<b>;</b>	<b>STOP</b>
<b>;</b> Daten, s. Schritt 1				

**Schritt 3:** Lesen des aktuellen Elements und Befehlsmodifikation

<b>;</b> FOR-Schleife (Teil 1), s. Schritt 2				
<b>M10:</b>	<b>LDA</b>	<b>ZAHL</b>	<b>;</b>	<b>A := &lt;ZAHL&gt;</b>
<b>;</b> Verarbeiten des aktuellen Elements				
	<b>LDA</b>	<b>M10</b>	<b>;</b>	<b>A := &lt;M10&gt;, Befehlsmodifikation</b>
	<b>LDB</b>	<b>EINS</b>	<b>;</b>	<b>B := 1</b>
	<b>ADD</b>		<b>;</b>	<b>A := A + B</b>
	<b>MOV</b>	<b>M10</b>	<b>;</b>	<b>&lt;M10&gt; := A</b>
<b>;</b> FOR-Schleife (Teil 2), s. Schritt 2				
<b>;</b> Daten, s. Schritt 1				

#### Schritt 4:           Verarbeiten des aktuellen Elements

```
; FOR-Schleife (Teil 1), s. Schritt 2  
  
; Lesen des aktuellen Elements, s. Schritt 3  
  
    LDB   SUMME   ; B := <SUMME>  
    ADD           ; A := A + B  
    MOV   SUMME   ; <SUMME> := A  
  
; Befehlsmodifikation, s. Schritt 3  
  
; FOR-Schleife (Teil 2), s. Schritt 2  
  
; Daten, s. Schritt 1
```

Damit ist das Programm komplett:

```
M1:    LDA    ANZ       ; A := <ANZ>  
      LDB    EINS       ; B := 1  
      SUB           ; A := A - B  
      MOV    ANZ       ; <ANZ> := A  
      JPS    M99       ; IF A < 0 GOTO M9  
;  
M10:   LDA    ZAHL       ; A := <ZAHL>  
;  
      LDB    SUMME    ; B := <SUMME>  
      ADD           ; A := A + B  
      MOV    SUMME    ; <SUMME> := A  
;  
      LDA    M10       ; A := <M10>  
      LDB    EINS       ; B := 1  
      ADD           ; A := A + B  
      MOV    M10       ; <M10> := A  
;  
      JMP    M1       ; GOTO M1  
M99:   HLT           ; STOP  
;  
EINS:   DEF    1       ; KONSTANTE 1  
;  
SUMME:   DEF   0       ; SUMME  
;  
ANZ:    DEF    5       ; ANZAHL  
ZAHL:   DEF    4       ; ZAHLENFELD  
      DEF   -8       ;  
      DEF   11       ;  
      DEF   39       ;  
      DEF   54       ;
```

Problemabhängig ist hier nur Schritt 4.

Ich gebe noch einige Beispiele an. Weitere finden Sie in Übung 4.

- Wieviele Zahlen des Feldes sind negativ?

	JPS	M11
	JMP	M12
M11:	LDA	NEG
	LDB	EINS
	ADD	
	MOV	NEG
M12:	...	
NEG:	DEF	0

- Wieviele Zahlen des Feldes sind gerade Zahlen ?

	RRA	1
	JPS	M11
	LDA	GERADE
	LDB	EINS
	ADD	
	MOV	GERADE
M11:	...	
GERADE:	DEF	0

Variante 1

	LDB	EINS
	AND	
	SUB	
	JPS	M11
	JMP	M12
M11:	LDA	GERADE
	ADD	
	MOV	GERADE
M12:	...	
GERADE:	DEF	0

Variante 2