

Übung 1: Funktionen, Prozeduren und Methoden ohne Rückgabewert

- Schreiben Sie eine Prozedur in Pseudocode, die alle Elemente eines Feldes um jeweils eine Position nach rechts verschiebt. Das letzte Element soll das erste Element werden.

aufrufendes Programm <pre>... integer feld[n]; ... //Feld mit Werten belegen verschiebe(feld); ...</pre>	<pre>procedure verschiebe(feld[]) begin n := Länge des Feldes feld; merke := feld[n-1]; for i := n-1 to 1 step -1 // Schrittweite -1 do feld[i] := feld[i-1]; end do feld[0] := merke; end</pre>
--	--

- Erstellen Sie eine Prozedur in Pseudocode, der als Parameter der Radius einer Kugel übergeben wird. In der Prozedur sind der Umfang der Kugel, die Mantelfläche und das Volumen zu berechnen. Die Werte sollen anschließend dem aufrufenden Programm zur Verfügung stehen.

Parameterübergaben über Referenzen

aufrufendes Programm <pre>... eingabe(r); berechne_kugel(r, u, m, v); ...</pre>	<pre>procedure berechne_kugel(r, ref u, ref m, ref v) begin pi := 3.14; u := 2 * pi * r; m := 4 * pi * r * r; v := (4 / 3) * pi * r * r * r; end</pre>
--	---

Übung 2: Funktionen und Methoden mit Rückgabewert

- Schreiben Sie eine Funktion in Pseudocode, die das Minimum dreier verschiedener Werte bestimmt und an das Hauptprogramm zurückgibt.

aufrufendes Programm <pre>... integer z1, z2, z3, min; eingabe(z1, z2, z3); min := minimum(z1, z2, z3); ...</pre>	<pre>function minimum(x, y, z) begin if (x < y) and (x < z) then min := x; else if (y < x) and (y < z) then min := y; else min := z; end if end if return min; end</pre>
--	--

3. Erstellen Sie eine Funktion in Pseudocode, die die n-te Potenz einer Zahl x berechnet ($y = x^n$). Die Zahlen x und n sollen der Funktion als Parameter übergeben werden, wobei x und n natürliche Zahlen (0, 1, 2, 3 ...) sind. Das Ergebnis soll im Hauptprogramm ausgegeben werden.

aufrufendes Programm ... integer y, x, n; ...//y, x, n initialisieren y = berechne_potenz(x, n); ...//Ausgabe ...	function berechne_potenz(x, n) begin potenz = 1; for i := 1 to n do potenz = potenz * x; end do return potenz; end
---	---