

Lehrplan	möglicher Inhalt
Kennen von Datenstrukturen - einfache Datentypen	Zahl (Ganzzahl/integer, Dezimalzahl/real) Text (Zeichen/char, Zeichenkette/string) Wahrheitswert (boolean)
- strukturierte Datentypen	Feld/array Verbund/record Datei/file ? Objekt/class ?
- höhere Datenstrukturen	Stapel, Schlange, Baum?

### mögliche Aufgabentypen zur Kontrolle

Inhaltsbereich	Prozessbereich	Vorgabe / bereitgestellte Inhalte der Aufgabe	Auftrag / zu erzeugende Inhalte der Aufgabe
Inf+Daten	Mod+Impl	Datentyp	Beispiele für entsprechende Informationen nennen
		konkretes Beispiel einer Information (Vorname, Alter...)	passenden Datentyp od. Datenstruktur nennen/zuordnen
	Begr+Bew	Information + vorgeschlagener Datentyp	Zustimmung zur Zuordnung oder Ablehnung begründen
	Stru+Netz	Information+ vorgeschlagener Datentyp (Feld oder Verbund)	strukturieren des Feldes (Typ, Dimension) oder des Verbunds (Felder und Typen)
	Kom+Koo		
	Dar+Int	verbale Beschreibung eines strukturierten Datentyps (Feld, Verbund)	zeichnerische Darstellung
		zeichnerische Darstellung eines strukturierten Datentyps (Feld, Verbund)	Interpretation / verbale Beschreibung

### Beispielaufgaben → Lösung

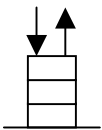
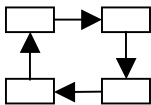
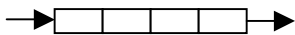
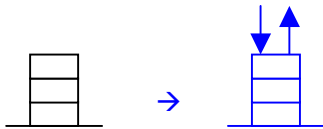
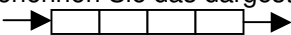
Beispielaufgaben → Lösung					BE															
1	Nennen Sie je ein typisches Beispiel für eine Information, die mit dem Datentyp „Zahl“, „Text“ bzw. „Wahrheitswert“ dargestellt werden kann.				3															
2	Ordnen Sie jeder Information den passenden Datentyp zu. <table><tr><th>Information</th><th>Datentyp</th></tr><tr><td>Bezeichnung: Schraube</td><td>→ Text</td></tr><tr><td>Länge in mm: 35</td><td>→ Zahl (Integer)</td></tr><tr><td>Preis in €: 0,02</td><td>→ Zahl (Real)</td></tr><tr><td>gehärtet: ja</td><td>→ Wahrheitswert</td></tr></table>				Information	Datentyp	Bezeichnung: Schraube	→ Text	Länge in mm: 35	→ Zahl (Integer)	Preis in €: 0,02	→ Zahl (Real)	gehärtet: ja	→ Wahrheitswert	4					
Information	Datentyp																			
Bezeichnung: Schraube	→ Text																			
Länge in mm: 35	→ Zahl (Integer)																			
Preis in €: 0,02	→ Zahl (Real)																			
gehärtet: ja	→ Wahrheitswert																			
3	Eine Wetterstation möchte die Tageshöchsttemperaturen eines Monats für die spätere Verarbeitung speichern. Für den Datentyp existieren zwei Vorschläge: A: Speicherung als Feld von 31 Integerwerten B: Speicherung als Verbund mit den Integerwerten Tag1, Tag2, Tag3, Tag4 ... Tag31 Entscheiden Sie, welcher Datentyp besser geeignet ist (mit Begründung). → Feld, da alle Daten den gleichen Typ aufweisen, leichte Adressierung durch den Index				2															
4	Für die Verwaltung eines Kleinbetriebes wird eine Datenstruktur benötigt, die die Informationen für einen Arbeiter (Name, Vorname, verheiratet, Straße, Postleitzahl, Ort, Gehalt) zusammenfasst. Schlagen Sie eine passende Datenstruktur vor und begründen Sie Ihre Wahl kurz. → Verbund, da unterschiedliche Datentypen zusammengefasst werden (Text, Wahrheitswert, Zahl)				2															
5	Um die Ansteuerung einer Ampel zu realisieren, soll der Zustand der roten, gelben und grünen Lampe in einem Feld aus 3 Wahrheitswerten gespeichert werden. Stellen Sie die Datenstruktur zeichnerisch dar. Belegen Sie das Feld so mit Werten, dass die Ampel das Signal „Rot-Gelb“ zeigt. → <table><tr><td>true</td><td>true</td><td>false</td></tr><tr><td>rot (an)</td><td>gelb (an)</td><td>grün (aus)</td></tr></table>				true	true	false	rot (an)	gelb (an)	grün (aus)	2									
true	true	false																		
rot (an)	gelb (an)	grün (aus)																		
6	Eine Materialdatenbank verwendet folgende Datenstruktur zur Beschreibung der Materialien: <table><tr><td>Bezeichnung</td><td>Durchmesser (mm)</td><td>Länge (m)</td><td>Masse (g)</td><td>rostfrei</td></tr><tr><td>Rohr</td><td>25</td><td>3,00</td><td>1840</td><td>false</td></tr></table> Interpretieren Sie den folgenden Datensatz. <table><tr><td>Rohr</td><td>25</td><td>3,00</td><td>1840</td><td>false</td></tr></table> → Rohr mit 25mm Durchmesser und 3m Länge, Masse 1840g, nicht rostfrei				Bezeichnung	Durchmesser (mm)	Länge (m)	Masse (g)	rostfrei	Rohr	25	3,00	1840	false	Rohr	25	3,00	1840	false	4
Bezeichnung	Durchmesser (mm)	Länge (m)	Masse (g)	rostfrei																
Rohr	25	3,00	1840	false																
Rohr	25	3,00	1840	false																

Lehrplan	möglicher Inhalt
Einblick gewinnen in Verarbeitungsprinzipien - LIFO, FIFO	typische Anwendungsfälle, animierte Darstellungen

**mögliche Aufgabentypen zur Kontrolle**

Inhaltsbereich	Prozessbereich	Vorgabe / bereitgestellte Inhalte der Aufgabe	Auftrag / zu erzeugende Inhalte der Aufgabe
Inf+Daten	Mod+Impl	typisches Problem der Datenverarbeitung (Warteschlange, ...)	zuordnen des Verarbeitungsprinzips
		Verarbeitungsprinzip	zuordnen eines Anwendungsbeispiels
		Verarbeitungsprinzip	Bezeichnung erläutern
	Begr+Bew	typisches Problem der Datenverarbeitung (Warteschlange, ...) + vorgeschlagenes Verarbeitungsprinzip	Zustimmung zur Zuordnung oder Ablehnung begründen
	Stru+Netz		
	Kom+Koo		
	Dar+Int	Verarbeitungsprinzip + mehrere zeichnerische Darstellungen	zeichnerische Darstellung zuordnen
		Verarbeitungsprinzip + unvollständige zeichnerische Darstellung	zeichnerische Darstellung ergänzen (nicht produzieren)
		zeichnerische Darstellung eines Verarbeitungsprinzips	Verarbeitungsprinzip benennen

**Beispielaufgaben → Lösung**

		BE
1	Ein Platzkarten-Reservierungssystem soll alle über das Internet eingehenden Reservierungswünsche speichern und in der gleichen Reihenfolge abarbeiten. Benennen Sie das erforderliche Verarbeitungsprinzip (Abkürzung und Übersetzung) → FIFO, was zuerst eingeht, wird zuerst ausgegeben (bearbeitet)	2
2	Beschreiben Sie kurz einen typischen Anwendungsfall für das LIFO-Prinzip → Die Bearbeiten-Rückgängig-Funktion von Anwendungsprogrammen macht die letzte Aktion zuerst rückgängig.	1
3	Beim Telefonieren werden die Ziffern der Rufnummer nacheinander eingegeben und anschließend zum Herstellen der Verbindung verwendet. Entscheiden Sie begründet, ob bei der Verarbeitung der Ziffernfolge das FIFO- oder das LIFO-Prinzip verwendet wird. → FIFO, denn die zuerst eingegebene Ziffer wird zuerst weiterverwendet.	2
4	Alle für einen Flugzeugstart nötigen Aktionen (Türen verriegeln, Motoren anschalten, ...) sollen in der richtigen Reihenfolge im Bordcomputer gespeichert und bei Bedarf automatisch ausgeführt werden. Der Programmierer schlägt vor, das LIFO-Prinzip einzusetzen. Begründen Sie Ihre Zustimmung bzw. Ablehnung zu diesem Vorschlag. → Ablehnung, da die Aktionen sonst in der entgegengesetzten Reihenfolge ausgeführt werden (oder in der entgegengesetzten Reihenfolge gespeichert werden müssten).	2
5	Ordnen Sie dem FIFO- und dem LIFO-Prinzip die passende Darstellung zu. A  B  C  → FIFO=C, LIFO=A	2
6	Ergänzen Sie die Pfeile für die Eingabe und Ausgabe so, dass das LIFO-Prinzip dargestellt wird. 	1
7	Benennen Sie das dargestellte Verarbeitungsprinzip und die zugrunde liegende Datenstruktur. →  → FIFO-Prinzip, Datenstruktur: Schlange	2

Lehrplan	möglicher Inhalt
Beherrschen der Implementierung ausgewählter Datenstrukturen in einer Programmiersprache	<p>einfach: Zahl (Ganzzahl, Dezimalzahl) Text (Zeichen?, Zeichenkette) Wahrheitswert (boolean)</p> <p>strukturiert: Feld Verbund?</p> <p>Wertzuweisungen eine Art der Wiederholung (z.B. nur i:=1; while ort[i]&lt;&gt;'Dresden' do i:=i+1; writeln(i); zum Suchen in einem Feld) Sonderfälle (Suchwert wird nicht gefunden) hier noch nicht behandeln</p>

mögliche Aufgabentypen zur Kontrolle			
Inhaltsbereich	Prozessbereich	Vorgabe / bereitgestellte Inhalte der Aufgabe	Auftrag / zu erzeugende Inhalte der Aufgabe
Spr+Auto	Mod+Impl	Variablenname+Datentyp	Deklaration als Quelltext (auf Papier/im Programm)
		Deklaration einer Variable	Name und Typ der Variable angeben
		Startbelegung von Variablen und einige Wertzuweisungen	Belegung der Variablen nach jeder Wertzuweisung
		Variablendeklaration und Formel (Volumen, Flächeninhalt o.ä.)	Quelltext für Berechnung / Wertzuweisungen
		Deklaration eines Feldes + Index + Wert, der dort gesetzt werden soll	Quelltext für Wertzuweisung
		Deklaration eines Feldes + gesuchter Wert + unvollständiger Quelltext (ohne Bedingung)	Quelltext mit der Bedingung vervollständigen
	Begr+Bew	Deklaration mit unpassendem Variablentyp (z.B. Kinderanzahl : string, verheiratet : integer)	begründen, dass der Typ unpassend ist und Verbesserungsvorschlag angeben
		Formel und vorgeschlagener Quelltext	begründen, warum die Wertzuweisung nicht korrekt ist und Verbesserungsvorschlag angeben
		Deklaration von Variablen und Wertzuweisung mit falschem Datentyp (z.B. Kinderzahl : integer; Kinderanzahl := '2' verheiratet := 'ja' statt := true)	begründen, warum die Wertzuweisung nicht korrekt ist und Verbesserungsvorschlag angeben
		Deklaration von Variablen und Wertzuweisungen mit verschiedenen Datentypen (z.B. Kinderanzahl :integer; Kinderzahl := '2' und ... :=2,)	auswählen der richtigen Variante und begründen
		Start mit a=2, b=3 Anweisungen: a:=b; b:=a;	begründen, warum diese Anweisungen nicht dazu führen, dass die Werte von a und b ausgetauscht werden
	Stru+Netz	Startbelegung von Variablen + geplante Endbelegung + einige Wertzuweisungen (drei bis vier)	Wertzuweisungen in die richtige Reihenfolge bringen, damit die geplante Endbelegung eintritt
	Kom+Koo		
	Dar+Int	Startbelegung von Variablen, einige Wertzuweisungen und unvollständige zeichnerische Darstellung	Darstellung der Wertzuweisung durch Pfeile, Zahlen u.s.w. vervollständigen

Beispielaufgaben → Lösung		BE																								
1	Notieren Sie die in Pascal erforderliche Deklaration folgender Variablen: Wohnort (Text), Alter (ganze Zahl), IstVegetarier (Wahrheitswert) → var Wohnort : string; Alter : integer; IstVegetarier : boolean; (1BE auf Syntax)	4																								
2	Für eine Wohnungsverwaltung wurde ein Programm in Pascal entwickelt, das unter anderem die folgenden Variablen deklariert. var Postleitzahl : string; Stockwerk : integer; Internetanschluss : boolean; Zimmer : real; Ordnen Sie in der Tabelle den Datentypen die Bezeichnungen der Variablen zu. <table><tr><th>Datentyp</th><th>Bezeichnung der Variablen</th></tr><tr><td>Zahl</td><td>→ Stockwerk, Zimmer</td></tr><tr><td>Wahrheitswert</td><td>→ Internetanschluss</td></tr><tr><td>Text</td><td>→ Postleitzahl</td></tr></table>	Datentyp	Bezeichnung der Variablen	Zahl	→ Stockwerk, Zimmer	Wahrheitswert	→ Internetanschluss	Text	→ Postleitzahl	4																
Datentyp	Bezeichnung der Variablen																									
Zahl	→ Stockwerk, Zimmer																									
Wahrheitswert	→ Internetanschluss																									
Text	→ Postleitzahl																									
3	In einem Pascalprogramm werden die folgenden Anweisungen ausgeführt. Ergänzen Sie in jeder Zeile die Werte der Variablen nach Ausführung der Anweisung. <table><tr><th>Startbelegung</th><th>A</th><th>B</th><th>C</th></tr><tr><td></td><td>1</td><td>5</td><td>4</td></tr><tr><td>A := 12 / C;</td><td>→ 3</td><td></td><td></td></tr><tr><td>B := C - A;</td><td></td><td>→ 1</td><td></td></tr><tr><td>C := C + 1;</td><td></td><td></td><td>→ 5</td></tr><tr><td>B := A ;</td><td></td><td>→ 3</td><td></td></tr></table>	Startbelegung	A	B	C		1	5	4	A := 12 / C;	→ 3			B := C - A;		→ 1		C := C + 1;			→ 5	B := A ;		→ 3		4
Startbelegung	A	B	C																							
	1	5	4																							
A := 12 / C;	→ 3																									
B := C - A;		→ 1																								
C := C + 1;			→ 5																							
B := A ;		→ 3																								
4	In einem Pascal-Programm zur Dreiecksberechnung sind die folgenden Variablen deklariert. var a,b, gamma_Grad, gamma_Bogen, Flaeche : real; a) Aus dem im Gradmaß gegebenem Winkel Gamma soll der Winkel im Bogenmaß nach der Formel $\text{Bogenmaß} = \text{Gradmaß} \cdot \frac{3,14}{180}$ berechnet werden. Notieren Sie die Anweisung. → gamma_Bogen := gamma_Grad * 3,14 / 180 b) Aus den Seitenlängen (a, b) und dem Winkel γ im Bogenmaß soll der Flächeninhalt des Dreiecks nach der Formel $A = \frac{1}{2} \cdot a \cdot b \cdot \sin \gamma$ berechnet werden. Notieren Sie die Anweisung. → Flaeche := a * b * sin( gamma_Bogen ) / 2 oder 0.5*a*b* sin( gamma_Bogen )	4																								
5	Die Gehälter der 97 Mitarbeiter einer Firma sind in einem Feld gespeichert, das wie folgt deklariert ist. var Gehalt : array [1..100] of real; Notieren Sie die Anweisung, mit der für den zwölften Mitarbeiter das Gehalt auf 850 € festgelegt wird. → Gehalt[12] := 850 (Index, Wert und Typ des Gehaltes müssen stimmen, also nicht 850€!)	3																								
6	Eine Firma speichert für jeden Außendienstmitarbeiter den Ort, an dem er gerade tätig ist in dem Feld Ort (siehe Deklaration). Ergänzen Sie den Quelltext so, dass die Nummer des Mitarbeiters, der in Heidenau tätig ist, am Ende in der Variablen n gespeichert ist. var Ort : array [1..500] of string; i, n : integer; begin n := 0; i := 1; while i <= 500 do begin if ..... then ..... i := i + 1; end; end; → Ort[i] = 'Heidenau' → n := i;	3																								

7	<p>Für die Familienkasse wird ein neues Programm zur Berechnung des Kindergeldes geplant. Die Eltern sollen für jedes Kind 80,50€ Kindergeld erhalten. Der Programmierer schlägt vor:</p> <pre>var Kinder : string; Kindergeld : integer; begin   if Kinder &gt; '0' then     Kindergeld := Kinder * 80,50 ;   end;</pre> <p>Entscheiden Sie (mit Begründung) ob die verwendeten Datentypen geeignet sind. Korrigieren Sie gegebenenfalls die Datentypen.</p> <p>→ beide Typen sind ungeeignet; mit der Kinderzahl muss gerechnet werden (Typ Zahl), das Kindergeld ist keine ganze Zahl. Korrektur: var Kinder : integer; Kindergeld : real;</p>	4																								
8	<p>In einem Pascal-Programm soll der Flächeninhalt eines Trapezes nach der Formel</p> $A = \frac{a+c}{2} \cdot h$ <p>berechnet werden. Entscheiden Sie, ob die folgende Anweisung dazu geeignet ist (mit Begründung). Korrigieren Sie gegebenenfalls die Anweisung.</p> <pre>A := a + c / 2 * h</pre> <p>→ falsche Anweisung, da nur c halbiert wird. Korrektur: A := (a+c) / 2 * h</p>	2																								
9	<p>Für das Einwohnermeldeamt soll ein neues Verwaltungsprogramm entworfen werden, das unter anderem folgende Variablen deklariert.</p> <pre>var Vorname : string; verheiratet : boolean;     Postleitzahl : string; Etage : integer;</pre> <p>Prüfen und korrigieren Sie gegebenenfalls die verwendeten Wertzuweisungen.</p> <table><tr><th>Anweisung</th><th>korrekt? ja/nein</th><th>Korrektur</th></tr><tr><td>Vorname := 'Otto';</td><td>→ ja</td><td>-</td></tr><tr><td>verheiratet := 'ja';</td><td>→ nein</td><td>verheiratet := true;</td></tr><tr><td>Postleitzahl := 01238;</td><td>→ nein</td><td>Postleitzahl := '01238';</td></tr><tr><td>Etage := Erdgeschoss;</td><td>→ nein</td><td>Etage := 0;</td></tr></table>	Anweisung	korrekt? ja/nein	Korrektur	Vorname := 'Otto';	→ ja	-	verheiratet := 'ja';	→ nein	verheiratet := true;	Postleitzahl := 01238;	→ nein	Postleitzahl := '01238';	Etage := Erdgeschoss;	→ nein	Etage := 0;	4									
Anweisung	korrekt? ja/nein	Korrektur																								
Vorname := 'Otto';	→ ja	-																								
verheiratet := 'ja';	→ nein	verheiratet := true;																								
Postleitzahl := 01238;	→ nein	Postleitzahl := '01238';																								
Etage := Erdgeschoss;	→ nein	Etage := 0;																								
10	<p>Für das Einwohnermeldeamt soll ein neues Verwaltungsprogramm entworfen werden, das unter anderem folgende Variablen deklariert.</p> <pre>var Vorname : string; verheiratet : boolean;     Postleitzahl : string; Etage : integer;</pre> <p>Entscheiden Sie, welche Anweisung korrekt notiert ist.</p> <table><tr><th>Anweisung 1</th><th>Anweisung 2</th><th>Anweisung 3</th><th>Nr.</th></tr><tr><td>Otto := 'Vorname';</td><td>Vorname := 'Otto';</td><td>Vorname := Otto;</td><td>→2</td></tr><tr><td>verheiratet := 'true';</td><td>verheiratet := 'ja';</td><td>verheiratet := true;</td><td>→3</td></tr><tr><td>Postleitzahl := '012 38';</td><td>Postleitzahl := '01238';</td><td>Postleitzahl := 01238;</td><td>→2</td></tr><tr><td>Etage := 'Erdgeschoss';</td><td>Etage := Erdgeschoss;</td><td>Etage := 0;</td><td>→3</td></tr></table>	Anweisung 1	Anweisung 2	Anweisung 3	Nr.	Otto := 'Vorname';	Vorname := 'Otto';	Vorname := Otto;	→2	verheiratet := 'true';	verheiratet := 'ja';	verheiratet := true;	→3	Postleitzahl := '012 38';	Postleitzahl := '01238';	Postleitzahl := 01238;	→2	Etage := 'Erdgeschoss';	Etage := Erdgeschoss;	Etage := 0;	→3	4				
Anweisung 1	Anweisung 2	Anweisung 3	Nr.																							
Otto := 'Vorname';	Vorname := 'Otto';	Vorname := Otto;	→2																							
verheiratet := 'true';	verheiratet := 'ja';	verheiratet := true;	→3																							
Postleitzahl := '012 38';	Postleitzahl := '01238';	Postleitzahl := 01238;	→2																							
Etage := 'Erdgeschoss';	Etage := Erdgeschoss;	Etage := 0;	→3																							
11	<p>Innerhalb eines Programms sollen die Werte der Variablen a und b miteinander vertauscht werden.</p> <p>Entscheiden Sie (mit Begründung) ob die Anweisungsfolge</p> <pre>a := b; b := a;</pre> <p>dazu geeignet ist.</p> <p>→ Nein, die Anweisungsfolge ist ungeeignet, da nach a:=b der ursprüngliche Wert von a nicht mehr verfügbar ist.</p>	2																								
12	<p>In einem Pascalprogramm sollen die folgenden Anweisungen ausgeführt werden.</p> <pre>A := 15 / C;      B := C - A;      C := C + 1;</pre> <p>Bringen Sie diese Anweisungen in die richtige Reihenfolge, damit sich aus der Startbelegung die Endbelegung ergibt.</p> <table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>Startbelegung</td><td>1</td><td>7</td><td>4</td></tr><tr><td>C := C + 1;</td><td></td><td></td><td>→ 5</td></tr><tr><td>A := 15 / C;</td><td>→ 3</td><td></td><td></td></tr><tr><td>B := C - A;</td><td></td><td>→ 2</td><td></td></tr><tr><td>Endbelegung</td><td>3</td><td>2</td><td>5</td></tr></table> <p>Diese Knobelaufgabe ist eher als Zusatzaufgabe oder Hausaufgabe denkbar.</p>		A	B	C	Startbelegung	1	7	4	C := C + 1;			→ 5	A := 15 / C;	→ 3			B := C - A;		→ 2		Endbelegung	3	2	5	+1
	A	B	C																							
Startbelegung	1	7	4																							
C := C + 1;			→ 5																							
A := 15 / C;	→ 3																									
B := C - A;		→ 2																								
Endbelegung	3	2	5																							

<b>13</b>	<p>In einem Pascalprogramm werden die folgenden Anweisungen ausgeführt.</p> <pre> A := 12 / C; B := C - A; C := C + 1; </pre> <p>Stellen Sie die Wertzuweisungen durch beschriftete Pfeile dar und tragen Sie die neuen Werte der Variablen ein.</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <p><b>A</b></p> <div style="border: 1px solid black; padding: 2px; width: 60px; margin: 5px auto;">1</div> </div> <div style="text-align: center;"> <p><b>B</b></p> <div style="border: 1px solid black; padding: 2px; width: 60px; margin: 5px auto;">5</div> </div> <div style="text-align: center;"> <p><b>C</b></p> <div style="border: 1px solid black; padding: 2px; width: 60px; margin: 5px auto;">4</div> </div> </div> <p>→</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <p><b>A</b></p> <div style="border: 1px solid black; padding: 2px; width: 60px; margin: 5px auto;"> <del>1</del> 3 </div> </div> <div style="text-align: center;"> <p><b>B</b></p> <div style="border: 1px solid black; padding: 2px; width: 60px; margin: 5px auto;"> <del>5</del> 1 </div> </div> <div style="text-align: center;"> <p><b>C</b></p> <div style="border: 1px solid black; padding: 2px; width: 60px; margin: 5px auto;"> <del>4</del> 5 </div> </div> </div>	<b>3</b>
-----------	--	----------

Lehrplan	möglicher Inhalt
Beherrschen der Arbeit mit Unterprogrammen - Struktur von Unterprogrammen - Verwendung von Parametern	- formale und aktuelle Parameter - nur Wertparameter (aus Zeitgründen keine Referenzparameter)

mögliche Aufgabentypen zur Kontrolle			
Inhaltsbereich	Prozessbereich	Vorgabe / bereitgestellte Inhalte der Aufgabe	Auftrag / zu erzeugende Inhalte der Aufgabe
Spr+Auto	Mod+Impl	Formel (Volumen, Oberfläche o.ä.) + Name der Funktion	Deklarationsteil einer Funktion
		Deklarationsteil + Aufgabe der Prozedur + unvollständiger Quelltext	Quelltext mit wenigen Zeilen vervollständigen, dabei die formalen Parameter benutzen
	Begr+Bew	Deklarationsteil eines Unterprogramms	begründet entscheiden, ob es sich um eine Prozedur oder eine Funktion handelt
	Stru+Netz	Beschreibung eines Arbeitsablaufs, der sich wiederholende Elemente enthält	erkennen/nennen von Unterprogrammen, die wiederholt in diesem Ablauf auftreten (evtl. mit Angabe der formalen Parameter)
		Deklarationsteil eines Unterprogramms	Namen und Datentypen der formalen Parameter angeben
		Deklarationsteil eines Unterprogramms + verbale Beschreibung des aktuellen Problems	Quelltext für den Aufruf des Unterprogramms mit eingesetzten aktuellen Parametern
	Kom+Koo		
	Dar+Int	Quelltext eines Unterprogramms + zweimaliger Aufruf mit verschiedenen aktuellen Parametern	Eintragen der aktuellen Belegung der Variablen in jeder Programmzeile (zweifarbige für die beiden Durchläufe)

Beispielaufgaben → Lösung		BE
<b>1</b>	<p>Mit Hilfe einer selbst definierten Funktion „Flaeche“ soll der Flächeninhalt eines Trapezes berechnet werden. Zugrunde liegt die Formel <math>A = \frac{a+c}{2} \cdot h</math>.</p> <p>Notieren Sie den Deklarationsteil der Funktion unter Verwendung von geeigneten formalen Parametern.</p> <p>→ function Flaeche (a, b, h : real) : real;</p> <p>4 BE: Verwendung des Bezeichners Flaeche, Parameter, Typ der Parameter, Typ der Funktion</p>	<b>4</b>
<b>2</b>	<p>Mit Hilfe einer selbst definierten Funktion soll der Umfang eines Rechtecks berechnet werden. Zugrunde liegt die Formel <math>u = 2 \cdot (a + b)</math>. Vervollständigen Sie den Quelltext.</p> <pre> function Umfang_Rechteck (seite1, seite2 : real) : real; begin   ... → result := 2 * (seite1 + seite2)  oder  Umfang_Rechteck := 2 * (seite1 + seite2) end; </pre> <p>3 BE: Wertzuweisung, korrekter Term, Verwendung der formalen Parameter</p>	<b>3</b>

3	Entscheiden Sie (mit Begründung), ob mit der nachfolgenden Deklaration eine Funktion oder eine Prozedur vereinbart wurde. (Die Stelle XXXXX ist leider unleserlich.) XXXXXX Steuerbelastung ( Einnahmen, MwSt : real ) ; → Prozedur, da nach kein Datentyp für einen Rückgabewert deklariert ist	2																																																																																																																					
4	Ein Geldautomat durchläuft (vereinfacht) immer wieder den gleichen Zyklus. 1. PIN abfragen 2. PIN prüfen 3. Auszahlungsbetrag abfragen 4. Auszahlungsbetrag abzählen 5. Geld ausgeben Zur Realisierung der fünf Teilschritte sollen Unterprogramme eingesetzt werden. Deklarieren Sie für das 2. oder 4. Unterprogramm eine Funktion mit aussagekräftiger Bezeichnung, die einen Wahrheitswert zurückliefert (wahr bedeutet, dass die Prüfung erfolgreich verlaufen ist). Übergeben Sie die notwendigen Werte als formale Parameter. → 2. function PIN_korrekt (pin : integer) : boolean; → 4. function Betrag_korrekt (betrag : real) : boolean; 4 BE: Name, formaler Parameter, 2 Datentypen	4																																																																																																																					
5	In einem Programm der Familienkasse wird die folgende Funktion verwendet. function Kindergeld (verheiratet : boolean; Kinder : integer) : real; Vervollständigen Sie die Tabelle. <table><tr><td></td><td>Bezeichnung</td><td>Rückgabotyp bzw. Datentyp</td></tr><tr><td>Funktion</td><td>→ Kindergeld</td><td>→ Zahl (real)</td></tr><tr><td rowspan="2">formale Parameter</td><td>→ verheiratet</td><td>→ Wahrheitswert (boolean)</td></tr><tr><td>→ Kinder</td><td>→ Zahl (integer)</td></tr></table>		Bezeichnung	Rückgabotyp bzw. Datentyp	Funktion	→ Kindergeld	→ Zahl (real)	formale Parameter	→ verheiratet	→ Wahrheitswert (boolean)	→ Kinder	→ Zahl (integer)	6																																																																																																										
	Bezeichnung	Rückgabotyp bzw. Datentyp																																																																																																																					
Funktion	→ Kindergeld	→ Zahl (real)																																																																																																																					
formale Parameter	→ verheiratet	→ Wahrheitswert (boolean)																																																																																																																					
	→ Kinder	→ Zahl (integer)																																																																																																																					
6	Ein Programm für Architekten soll für beliebig große rechteckige Räume die Grundfläche (für den Fußbodenbelag) und die Fläche der Wände (zum Tapezieren) berechnen können. Es existiert eine Funktion für den Flächeninhalt eines Rechtecks. function Flaeche ( laenge, breite : real ) : real; //Angaben in Meter Länge, Breite und Höhe eines Raumes sind im Programm in den Variablen L, B und H gespeichert. var L, B, H, GF, WF: real; begin ... L := 5.8; B := 3.4; H := 2.3; Notieren Sie die Anweisungen, die dazu führen, dass ... a) ... die Variable GF die Grundfläche des Raumes enthält. b) ... die Variable WF die Wandfläche des Raumes enthält (ohne Decke). → a) GF := Flaeche ( L, B ); → b) WF := 2 * Flaeche ( L, H ) + 2 * Flaeche( B, H ); a) 1 BE: Einsetzen der aktuellen Parameter, b) 2 BE: aktuelle Parameter, Korrektheit des Terms	3																																																																																																																					
7	Die Funktion Max wird zweimal aufgerufen. Vervollständigen Sie die Belegungstabelle für beide Aufrufe. <table><tr><td></td><td colspan="4">Aufruf 1</td><td colspan="4">Aufruf 2</td></tr><tr><td></td><td>x</td><td>y</td><td>z</td><td>Max</td><td>x</td><td>y</td><td>z</td><td>Max</td></tr><tr><td>function Max ( x, y : real ) : real;</td><td>→ 3</td><td>→ 5</td><td></td><td></td><td>→ 9</td><td>→ 5</td><td></td><td></td></tr><tr><td>var z : real;</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>begin</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>z := x;</td><td></td><td></td><td>→ 3</td><td></td><td></td><td></td><td>→ 9</td><td></td></tr><tr><td>if y &gt; z then</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>z := x;</td><td></td><td></td><td>→ 5</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Max := z;</td><td></td><td></td><td></td><td>→ 5</td><td></td><td></td><td></td><td>→ 9</td></tr><tr><td>end;</td><td colspan="8"></td></tr><tr><td>...</td><td>a</td><td>b</td><td colspan="6"></td></tr><tr><td>a := Max ( 3, 5 ); // Aufruf 1</td><td>→ 5</td><td></td><td colspan="6"></td></tr><tr><td>b := Max ( 9, a ); // Aufruf 2</td><td></td><td>→ 9</td><td colspan="6"></td></tr></table> Diese Aufgabe ist eher als Übungsaufgabe zu verwenden, da sie viel Zeit und Konzentration erfordert.		Aufruf 1				Aufruf 2					x	y	z	Max	x	y	z	Max	function Max ( x, y : real ) : real;	→ 3	→ 5			→ 9	→ 5			var z : real;									begin									z := x;			→ 3				→ 9		if y > z then									z := x;			→ 5						Max := z;				→ 5				→ 9	end;									...	a	b							a := Max ( 3, 5 ); // Aufruf 1	→ 5								b := Max ( 9, a ); // Aufruf 2		→ 9							6
	Aufruf 1				Aufruf 2																																																																																																																		
	x	y	z	Max	x	y	z	Max																																																																																																															
function Max ( x, y : real ) : real;	→ 3	→ 5			→ 9	→ 5																																																																																																																	
var z : real;																																																																																																																							
begin																																																																																																																							
z := x;			→ 3				→ 9																																																																																																																
if y > z then																																																																																																																							
z := x;			→ 5																																																																																																																				
Max := z;				→ 5				→ 9																																																																																																															
end;																																																																																																																							
...	a	b																																																																																																																					
a := Max ( 3, 5 ); // Aufruf 1	→ 5																																																																																																																						
b := Max ( 9, a ); // Aufruf 2		→ 9																																																																																																																					