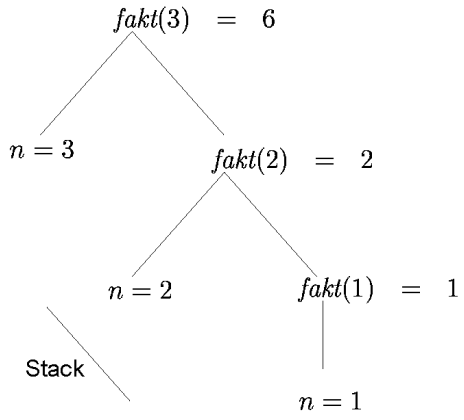
STOFF FÜR INFO DVP I (tabellarisch, ungeordnet)

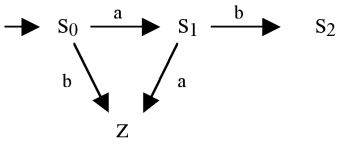
Syntax	Regeln für Aufbau der Wörter
funktionale Semantik	Funktionsbeschreibung
operationale Semantik	Abfolge der Berechnungsschritte
BNF	Backus Naur Form; Beschreibung der Syntax z.B. <zahl> ::= [ziffer{ziffer | 0}*] | 0 <ziffer> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
Alphabet	geordneter Zeichenvorrat
Darstellung eines **Algorithmus**	präzise Beschreibung der einzelnen Schritte eines **Verfahrens** endlich viele Schritte!
funktionale (applikative) Programmierung	nur Ausdrücke und Funktionen, keine Zuweisungen Rekursionen, Einbettung
Ablaufbaum	Graph für rekursiven Ablauf 
Signaturgraph	beschreibt (teilweise) eine Algebra. zeigt Syntax! Sorten und Funktionalitäten
Grundterme (zu Signatur)	Terme ohne freie Variablen, z.B. *add(zero, zero)*
Algebra / Rechenstruktur	Familie von Mengen und Abbildungen dazwischen
Boolsche Algebra	Algebra über den Wahrheitswerten
Landausymbole (Komplexität)	$f(n) \in O(g(n)) \Leftrightarrow \left\lvert \frac{f(n)}{g(n)} \right\rvert \leq M$ $f(n) \in o(g(n)) \Leftrightarrow \lim_{n \to \infty} \left\lvert \frac{f(n)}{g(n)} \right\rvert = 0$ gilt $f(n) \in O(g(n)) \wedge f(n) \notin o(g(n))$, dann heißt $O(g(n))$ die *kleinste Komplexität*

prozedurale (implikative, zuweisungsorientierte) Programmierung	**Anweisungen** Schleifen, Pointer, OoP
Äquivalenzklassenbildung	z.B. alle Zahlen, die bis auf die führenden Nullen gleich sind, werden zusammengefasst.
Backtracking	Algorithmus, der in einem endlichen Labyrinth einen Ausgang findet, wenn es einen gibt. Andernfalls stellt er fest, dass es keinen gibt.
call by value	Wert des Parameters wird übergeben (also kopiert)
call by name, Seiteneffekt	Verweis auf Parameter wird übergeben. -> Seiteneffekt
Sortierverfahren (für Reihen)	
Bubblesort	n mal die ganze Reihe durchlaufen und ggf. vertauschen! $O(n^2)$
Maxsort	Reihe von oben nach unten durchlaufen, Maximum berechnen, ggf. vertauschen! $O(n^2)$
Quicksort	- Zerlegen der Reihe in ein beliebiges Element und die Reihen rechts und links davon - suche linkes Element, das größer x und rechtes Element, das kleiner x und vertausche diese - wird nur ein Element gefunden, vertausche es mit x O(n*log n)
Mergesort	Aufteilung ähnlich wie Quicksort, Zusammenfügen in eine neue Liste (also keine Vertauschungen). -> zusätzlicher Speicherplatz benötigt!
Heapsort	Sortierverfahren mit Baummodell
Rekursionen	
einfach	lineare Komplexität
linear	rekursiver Aufruf in jedem Zweig höchstens einmal
kaskadenartig	in mindestens einem Zweig einer Fallunterscheidung zwei oder mehr rekursive Aufrufe
repititiv	rekursiver Aufruf in allen rekursiven Aufrufen als äußerste Rekursion, z.B. Fac

vernestet	rekursive Aufrufe als Parameter, z.B. func(func(x))
verschränkt	rekursive Funktionen rufen sich gegenseitig auf
rekursive Datenstruktur (in Gopher)	z.B. data Tree = Empty | Root Tree Tree
OoP	Klassen, Objekte, Vererbung, Polymorphie, Persistenz
Klasse / Schema	Signatur und Konstruktoren Besteht aus Feldern (Methoden, Variablen) konkret oder abstrakt
Kapselung	Felder sind an ihre Klasse gebunden. Stärke der Bindung siehe Integrität!
Konstruktoren	Methoden ohne Identifikator
Vererbung	die erbende Klasse erbt alle Variablen und Methoden, nicht aber die Konstruktoren!
Polymorphie (late binding)	verschiedene Methoden für Identifikator (unterscheidbar mit Parameterliste!)
Persistenz	Objekte existieren über deren Laufzeit hinweg.
Package	alle Klassen im Ordner
Integrität	Zugriffsrechte und Vererbungsrechte

Zugreifbar in	**public**	***default***	**protected**	**private**
Fremdklasse im selben package	ja	ja	ja	nein
Unterklasse im selben package	ja	ja	ja	nein
Fremdklasse in *anderem* package	ja	nein	nein	nein
Unterklasse in *anderem* package	ja	nein	nein	nein
geerbt				
Unterklasse im selben package	ja	ja	ja	nein
Unterklasse in *anderem* package	ja	nein	ja	nein

Interface	Klasse, bei der alle Methoden abstrakt sein müssen
Baumtypen	p-adisch, Binärbaum, geordneter BB, lineare Liste, AVL-Baum
vollständig ausgeglichener (Binärbaum)	alle Ebenen besetzt, außer der letzen
knotenausgeglichen	Knotenzahlen der Unterbäume unterscheiden sich max. um 1
höhenausgeglichen	Höhen der Unterbäume unterscheiden sich max. um 1

AVL-Baum	geordneter Binärbaum. in jedem Knoten gilt: $\lvert h(lUB) - h(rUB)\rvert \leq 1$ nach Löschen oder Einfügen kann die AVL-Eigenschaft durch Rotationen oder Doppelrotationen wieder hergestellt werden (mit O(log n)).
(endlicher) **Automat**	**Zustandsübergangssystem** (ZÜS) A = {S, I, δ, s_0, F} S : Zustandsmenge I : Alphabet δ : ZÜS-Funktion s_0 : Anfangszustand F : Endzustandsmenge endlich, weil Z endlich! nicht darstellbar: arithmetische Ausdrücke → Kellerautomat!
Fangzustand	bisher nicht angegebene Übergänge führen auf einen Fangzustand, z.B. 
Wortmenge	W(A) = {x ∈ I^* | s_0x ∈ F}
Pumping Lemma	Automat kann durch Zyklen auch unendliche Wortmengen akzeptieren
nicht-deterministisch	Übergang mit einem Eingabezeichen zu mehr als einem Folgezustand, z.B. δ(s_0, a) = s_0 oder s_1 δ ist Relation!
Myhill-Verfahren	zur Transformation eines nicht-deterministischen Automaten zu einem deterministischen (akzeptierte Wortmenge bleibt gleich!)
Minimalisierung	wenn δ(s, a) = δ(t, a) für jedes a ∈ I, dann kann man die beiden Zustände s und t zu einem zusammenfassen
äquivalent	2 Automaten haben die gleiche Wortmenge
formale Sprache	Werkzeug zur Darstellung von Teilmengen über Alphabet
formale Grammatik	G = {V, N, T, P, Z} V : Vokabular (V ⊆ Alphabet I) N : nicht-terminale Zeichen (dürfen im letzendlichen Wort nicht vorkommen (V \ I) T : terminale Teichen P : Produktionen Z : Startsymbol, **Axiom**

ableitbare Wortmenge	W = alle ableitbaren Zeichenreihen
Sprachschatz	$L = W \cap T^*$
Chomsky-3 Grammatik	alle Produktionen sind rechtslinear oder terminal $A \rightarrow a$ oder $A \rightarrow aB$
Chomsky-2 Grammatik	Produktionen von der Form $X \rightarrow x$ mit $x \in V^*$ ε-frei und kontextfrei! wortlängenmonoton (nach Ableiten ist Wortlänge nicht kürzer)
Automat → Grammatik	$V := S \cup I$ $N := S$ $T := I$ $P := \delta$ $Z := s_0$
Parsing	Umwandlung in eine Baumstruktur
linkskanonisch	beim Ableiten (mit Grammatik) immer zuerst den linken Teil weiter ableiten!
Kellerautomat	$K = \{S, I, T, b, s_0, Z, F\}$ T : endliches Kelleralphabet Z : Anfangszeichen im Keller

viel Glück!!!