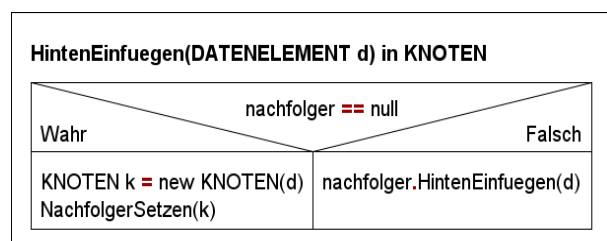
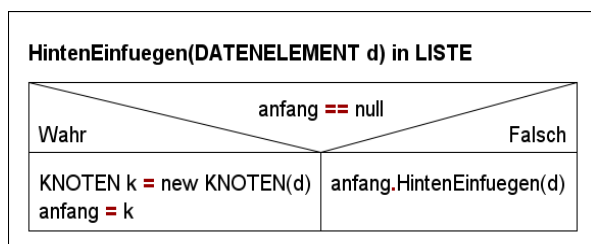
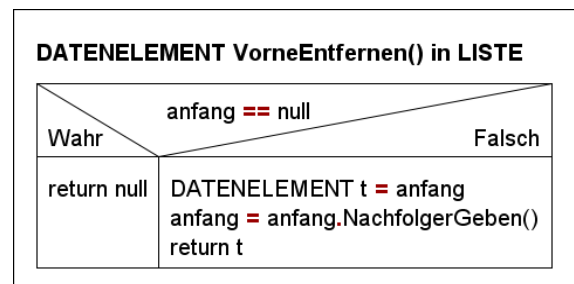
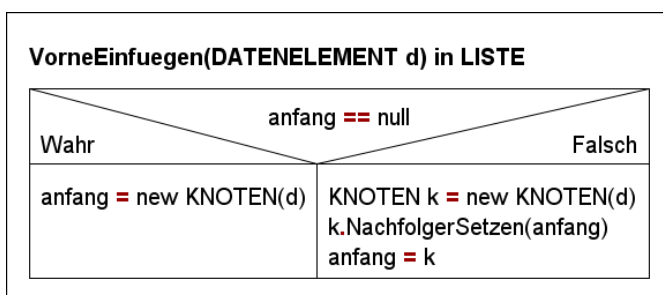


1.5 Die Klasse LISTE

Wir haben in der Zeit bisher gesehen, dass die LISTE an vielen Stellen verwendet werden kann. Nun geht es darum, die LISTE so abzuschließen, dass sie so optimal wie möglich funktioniert und in so vielen Anwendungen wie möglich wiederverwendbar ist. Um Speicher zu sparen, reduzieren wir die Referenzen auf den Anfang der LISTE. Dabei erweitern wir die LISTE um zwei Arten von Methoden. Zum einen ermöglichen wir das Einfügen und Entfernen an verschiedenen Stellen (Strukturmethoden), zum anderen fügen wir Methoden ein, die den Inhalt eines KNOTEN überprüfen. Das DATENELEMENT muss dementsprechend wie folgt erweitert werden:

LISTE	<<Schnittstelle>> DATENELEMENT
KNOTEN anfang	//keine!!
LISTE() void VorneEinfuegen(DATENELEMENT) void HintenEinfuegen(DATENELEMENT) void AnStelleEinfuegen(int) DATENELEMENT VorneEntfernen() DATENELEMENT HintenEntfernen() DATENELEMENT AnStelleEntfernen(int) void KnotenLoeschen(DATENELEMENT) void SortiertEinfuegen(DATENELEMENT) void VorEinfuegen(DATENEL.,DATENEL.) DATENELEMENT Suchen(DATENELEMENT) int LaengeGeben() void InformationAusgeben() void Darstellen()	void InformationAusgeben() void Darstellen(int) boolean SchluessellstGleich(DATENEL.) boolean IstKleinerAls(DATENELEMENT)

Überlegen wir uns nun, wie die Struktogramme der jeweiligen Methoden aussehen:



DATENELEMENT HintenEntfernen() in LISTE	
Wahr	Falsch
anfang.NachfolgerGeben() == null	
DATENELEMENT t = anfang anfang = null return t	return anfang.HintenEntfernen()

DATENELEMENT HintenEntfernen() in KNOTEN	
Wahr	Falsch
nachfolger.NachfolgerGeben() == null	
DATENELEMENT t = nachfolger NachfolgerSetzen(null) return t	return nachfolger.HintenEntfernen()

KnotenLoeschen(DATENELEMENT d) in KNOTEN		
Wahr	Falsch	
nachfolger != null		
Wahr	Falsch	//do nothing
nachfolger.DatenGeben().SchluesselstGleich(d)		
nachfolger = nachfolger.NachfolgerGeben()	nachfolger.KnotenLoeschen(d)	

KnotenLoeschen(DATENELEMENT d) in LISTE		
Wahr	Falsch	
anfang != null		
Wahr	Falsch	//do nothing
anfang.DatenGeben().SchluesselstGleich(d)		
anfang = anfang.NachfolgerGeben()	anfang.KnotenLoeschen(d)	

int LaengeGeben() in KNOTEN	
Wahr	Falsch
nachfolger == null	
return 0	return nachfolger.LaengeGeben()+1

int LaengeGeben() in LISTE	
Wahr	Falsch
anfang == null	
return 0	return anfang.LaengeGeben()+1

DATENELEMENT Suchen(DATENELEMENT d) in LISTE

	anfang == null		
Wahr			Falsch
return null	anfang.DatenGeben().SchluesselIstGleich(d)		Falsch
	Wahr	return anfang.DatenGeben()	Falsch
		return anfang.Suchen(d)	

DATENELEMENT Suchen(DATENELEMENT d) in KNOTEN

	nachfolger == null		
Wahr			Falsch
return null	nachfolger.DatenGeben().SchluesselIstGleich(d)		Falsch
	Wahr	return nachfolger.DatenGeben()	Falsch
		return nachfolger.Suchen(d)	

VorEinfuegen(DATENELEMENT wo, DATENELEMENT was) in LISTE

	anfang == null		
Wahr			Falsch
KNOTEN k = new KNOTEN(was) anfang = k	anfang.DatenGeben().SchluesselIstGleich(wo)		Falsch
	Wahr	KNOTEN k = new KNOTEN(was) k.NachfolgerSetzen(anfang) anfang = k	Falsch
		anfang.VorEinfuegen(wo, was)	

VorEinfuegen(DATENELEMENT wo, DATENELEMENT was) in KNOTEN

	nachfolger == null		
Wahr			Falsch
KNOTEN k = new KNOTEN(was) nachfolger = k	nachfolger.DatenGeben().SchluesselIstGleich(wo)		Falsch
	Wahr	KNOTEN k = new KNOTEN(was) k.NachfolgerSetzen(nachfolger) nachfolger = k	Falsch
		nachfolger.VorEinfuegen(wo, was)	

SortiertEinfuegen(DATENELEMENT d) in LISTE

	anfang == null		
Wahr			Falsch
KNOTEN k = new KNOTEN(d) anfang = k	anfang.DatenGeben().IstKleinerAls(d)		Falsch
	Wahr	anfang.SortiertEinfuegen(d)	Falsch
		KNOTEN k = new KNOTEN(d) k.NachfolgerSetzen(anfang) anfang = k	

SortiertEinfuegen(DATENELEMENT d) in KNOTEN

	nachfolger == null		
Wahr			Falsch
KNOTEN k = new KNOTEN(d) nachfolger = k	nachfolger.DatenGeben().IstKleinerAls(d)		
	Wahr		Falsch
	nachfolger.SortiertEinfuegen(d)	KNOTEN k = new KNOTEN(d) k.NachfolgerSetzen(nachfolger) nachfolger = k	

DATENELEMENT AnStelleEntfernen(int i) in KNOTEN

	nachfolger == null		
Wahr			Falsch
return null	i == 1		
	Wahr		Falsch
	DATENELEMENT t = nachfolger nachfolger = nachfolger.NachfolgerGeben() return t	return nachfolger.AnStelleEntfernen(i-1)	

DATENELEMENT AnStelleEntfernen(int i) in LISTE

	anfang == null && i >= 1		
Wahr			Falsch
return null	i == 1		
	Wahr		Falsch
	DATENELEMENT t = anfang anfang = anfang.NachfolgerGeben() return t	anfang.AnStelleEntfernen(i-1)	

AnStelleEinfuegen(DATENELEMENT d, int i) in LISTE

	anfang == null		
Wahr			Falsch
KNOTEN k = new KNOTEN(d) anfang = k	anfang.AnStelleEinfuegen(d, i-1)		

AnStelleEinfuegen(DATENELEMENT d, int i) in KNOTEN

nachfolger == null		
Wahr	Falsch	
KNOTEN k = new KNOTEN(d) nachfolger = k	i == 1	
	Wahr	Falsch
	KNOTEN k = new KNOTEN(d) k.NachfolgerSetzen(nachfolger) NachfolgerSetzen(k)	AnStelleEinfuegen(d, i-1)

Aufgaben:

- 1) Formuliere für alle neuen Methoden passende Struktogramme.
- 2) Setze die Methoden in den jeweiligen Klassen der Vorlage um.