I Erläuterungen

Voraussetzungen gemäß KCGO und Abiturerlass in der für den Abiturjahrgang geltenden Fassung

Standardbezug
Die nachfolgend ausgewiesenen prozessbezogenen Kompetenzbereiche sind für die Bearbeitung der jeweiligen Aufgabe besonders bedeutsam. Darüber hinaus können weitere, hier nicht ausgewiesene prozessbezogene Kompetenzbereiche für die Bearbeitung der Aufgabe nachrangig bedeutsam sein, zumal die Kompetenzbereiche in engem Bezug zueinander stehen. Die Operationalisierung des Standardbezugs erfolgt in Abschnitt II.

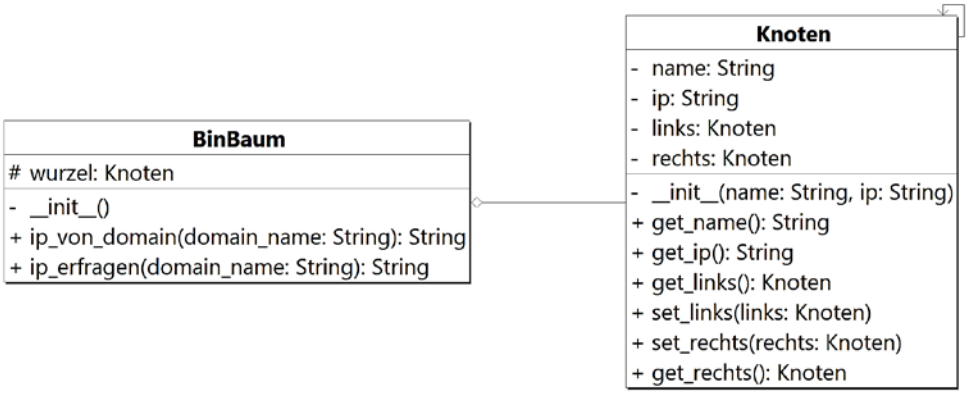
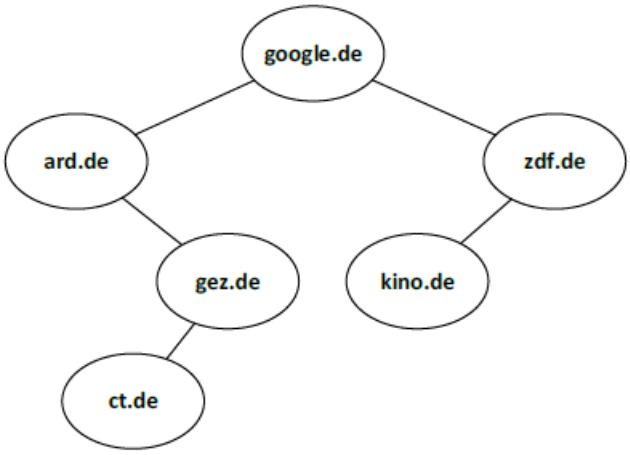
Aufgabe	Prozessbezogene Kompetenzbereiche				
	P1	P2	P3	P4	P5
1		X			
2				X	
3			X		
4				X	
5.1	X		X		
5.2			X		

Inhaltlicher Bezug
Der vorliegende Vorschlag bezieht sich schwerpunktmäßig auf die inhaltsbezogenen Kompetenzbereiche Algorithmen (I1) und Informatiksysteme (I4) nach KCGO.

Q1: Algorithmik und objektorientierte Modellierung
verbindliche Themenfelder: Such- und Sortieralgorithmen (Q1.1); Klassen und Objekte (Q1.3); Höhere Datenstrukturen und ihre objektorientierte Modellierung (Q1.4)

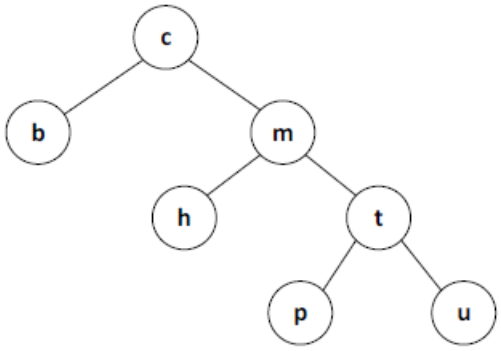
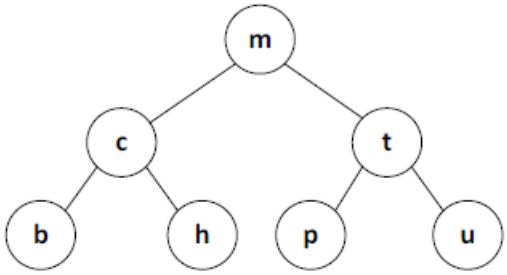
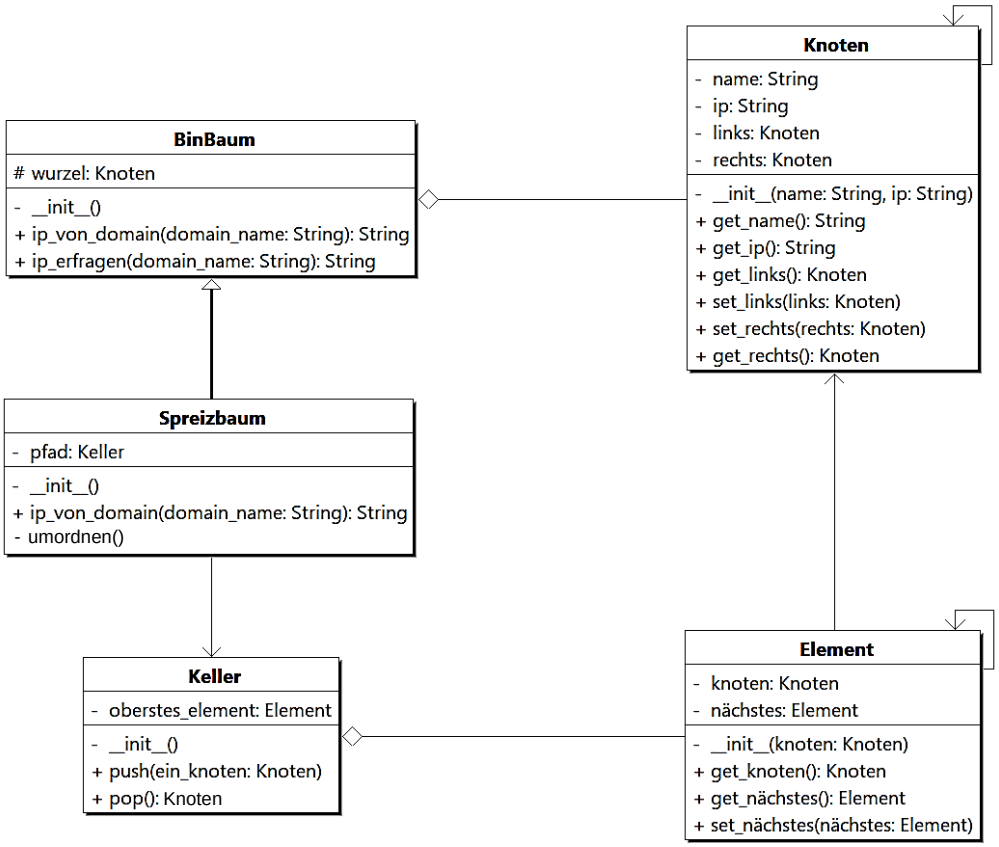
II Lösungshinweise und Bewertungsraster

In den nachfolgenden Lösungshinweisen sind alle wesentlichen Gesichtspunkte, die bei der Bearbeitung der einzelnen Aufgaben zu berücksichtigen sind, konkret genannt und diejenigen Lösungswege aufgezeigt, welche die Prüflinge erfahrungsgemäß einschlagen werden. Lösungswege, die von den vorgegebenen abweichen, aber als gleichwertig betrachtet werden können, sind ebenso zu akzeptieren.

<table>
<tr><th>Aufg.</th><th>erwartete Leistungen</th><th>BE</th></tr>
<tr><td>1</td><td> 
</td><td>5</td></tr>
<tr><td>2</td><td> 
</td><td>2</td></tr>
<tr><td>3</td><td><pre>
def ip_von_domain(self, domain_name: str) -> str:
    if self._wurzel is None:
        ip: str = self.ip_erfragen(domain_name)
        self._wurzel = Knoten(domain_name, ip)

    akt_knoten: Knoten = self._wurzel
    while domain_name != akt_knoten.get_name():
        if domain_name < akt_knoten.get_name():
            if akt_knoten.get_links() is None:
                ip: str = self.ip_erfragen(domain_name)
                ein_knoten: Knoten = Knoten(domain_name, ip)
                akt_knoten.set_links(ein_knoten)
            akt_knoten = akt_knoten.get_links()
        else:
            if akt_knoten.get_rechts() is None:
                ip: str = self.ip_erfragen(domain_name)
                ein_knoten: Knoten = Knoten(domain_name, ip)
                akt_knoten.set_rechts(ein_knoten)
            akt_knoten = akt_knoten.get_rechts()

    return akt_knoten.get_ip()
</pre></td><td>8</td></tr>
</table>

Aufg.	erwartete Leistungen	BE
4	Schritt 1  Schritt 2  	5
5.1	 	

Aufg.	erwartete Leistungen	BE
	Die Klasse Spreizbaum erbt als Unterklasse der Klasse BinBaum unter anderem die Methode ip_von_domain(domain_name). Allerdings muss diese Methode in der Klasse Spreizbaum anders implementiert werden als in der Klasse BinBaum, da für einen Spreizbaum nach jedem Suchvorgang die Methode umordnen() aufgerufen werden muss. Als Überschreiben einer geerbten Methode bezeichnet man das Implementieren einer Methode in einer Unterklasse mit dem gleichen Methodennamen und gleicher Signatur wie in der Oberklasse.	8

5.2	`def __umordnen_blatt(self):`
	`  kind: Knoten = self.__pfad.pop()`
	`  eltern: Knoten = self.__pfad.pop()`
	`  großeltern: Knoten = self.__pfad.pop()`
	
	`  if eltern.get_links() == kind:`
	`  eltern.set_links(None)`
	`  kind.set_rechts(eltern)`
	`  else:`
	`  eltern.set_rechts(None)`
	`  kind.set_links(eltern)`
	
	`  if großeltern.get_rechts() == eltern:`
	`  großeltern.set_rechts(kind)`
	`  else:`
	`  großeltern.set_links(kind)`
		7

	Summe	**35**

III Bewertung und Beurteilung

Die Bewertung und Beurteilung erfolgt unter Beachtung der nachfolgenden Vorgaben nach § 33 der Oberstufen- und Abiturverordnung (OAVO) in der jeweils geltenden Fassung. Bei der Bewertung und Beurteilung der sprachlichen Richtigkeit in der deutschen Sprache sind die Bestimmungen des § 9 Abs. 12 Satz 3 OAVO in Verbindung mit Anlage 9b anzuwenden.

Der Fehlerindex ist nach Anlage 9b zu § 9 Abs. 12 OAVO zu berechnen. Für die Ermittlung der Punkte nach Anlage 9a zu § 9 Abs. 12 OAVO bzw. des Abzugs nach Anlage 9b zu § 9 Abs. 12 OAVO wird jeweils der ganzzahlige nicht gerundete Prozentsatz bzw. Fehlerindex zugrunde gelegt. Der prozentuale sprachliche Anteil nach Anlage 9b zu § 9 Abs. 12 OAVO wird auf 20 % festgesetzt.

Darüber hinaus sind die Vorgaben der Erlasse „Hinweise zur Vorbereitung auf die schriftlichen Abiturprüfungen (Abiturerlass)“ und „Durchführungsbestimmungen zum Landesabitur“ in der für den Abiturjahrgang geltenden Fassung zu beachten.

Im Fach Informatik (Leistungskurs) werden Vorschläge zu den Themen der drei Kurshalbjahre Q1, Q2 und Q3 vorgelegt, wobei die Prüfungsleistung aus der Bearbeitung je eines Vorschlags zu jedem Halbjahresthema besteht, wofür insgesamt maximal 100 BE vergeben werden können. Ein Prüfungsergebnis von **5 Punkten (ausreichend)** setzt voraus, dass mindestens 45 % der zu vergebenden BE erreicht werden. Ein Prüfungsergebnis von **11 Punkten (gut)** setzt voraus, dass mindestens 75 % der zu vergebenden BE erreicht werden.

Gewichtung der Aufgaben und Zuordnung der Bewertungseinheiten zu den Anforderungsbereichen

Aufgabe	Bewertungseinheiten in den Anforderungsbereichen			Summe
	AFB I	AFB II	AFB III	
1	4	1		**5**
2	2			**2**
3		4	4	**8**
4		5		**5**
5.1	4	4		**8**
5.2		4	3	**7**
Summe	**10**	**18**	**7**	**35**

Die auf die Anforderungsbereiche verteilten Bewertungseinheiten innerhalb der Aufgaben sind als Richtwerte zu verstehen.