

I Erläuterungen

Aufgabenart

materialgebundene Aufgabenstellung

Voraussetzungen gemäß Lehrplan und Erlass „Hinweise zur Vorbereitung auf die schriftlichen Abiturprüfungen“ in der für den Abiturjahrgang geltenden Fassung

Q1: objektorientierte Modellierung

Objektmodell, Klassen, Beziehungen, UML-Klassendiagramm,

Listen, iterative Verfahren, Rekursion, Backtracking, Analyse von Algorithmen

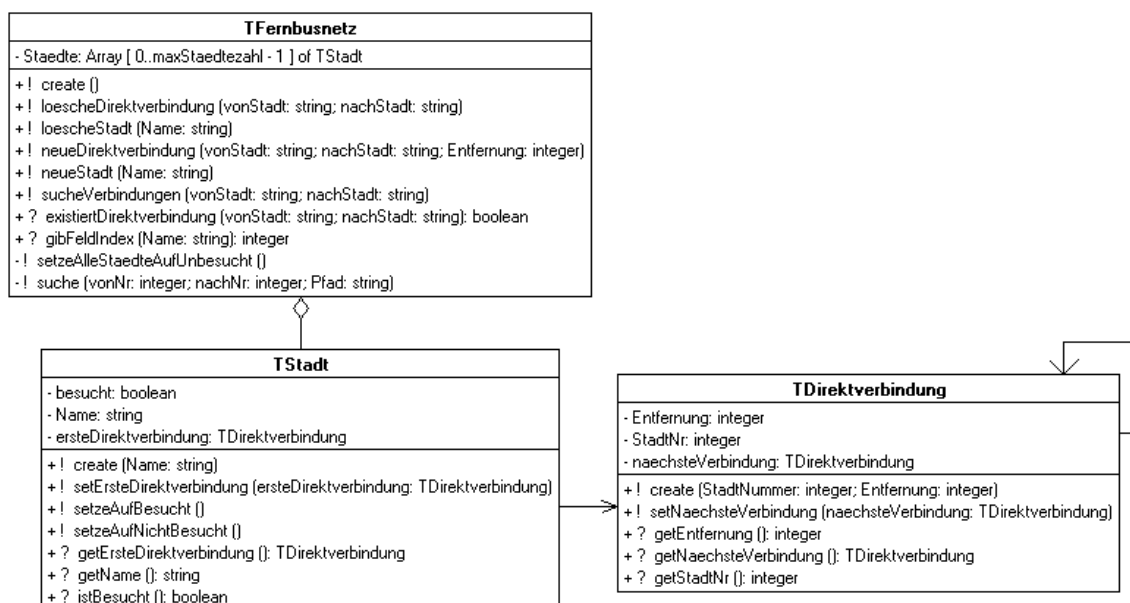
II Lösungshinweise und Bewertungsraster

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. Selbstverständlich sind jedoch Lösungswege, die von den vorgegebenen abweichen, aber als gleichwertig betrachtet werden können, ebenso zu akzeptieren.

Aufg.	erwartete Leistungen	BE
1.1	Die Städte werden in einem Feld der Größe 10 gespeichert. Jedes Feldelement besteht aus dem Namen der Stadt und einem Verweis auf die erste Direktverbindung. Zu jeder Stadt werden alle mit dieser Stadt direkt verbundenen Städte in Form einer linearen Liste gespeichert. Jedes Listenelement ist eine Direktverbindung, die die Stadtnummer, die Entfernung und einen Verweis auf das nächste Listenelement enthält.	5
1.2	In dem Feld <i>Staedte</i> der Klasse <i>TFernbusnetz</i> werden die vorhandenen Städte gespeichert. Es besteht daher zwischen der Klasse <i>TFernbusnetz</i> und der Klasse <i>TStadt</i> eine Aggregation. Eine Stadt kann eine Direktverbindung zu einer anderen Stadt haben (gerichtete Assoziation). Jede Direktverbindung, mit Ausnahme der letzten Direktverbindung, kann eine weitere Direktverbindung haben (rekursive Assoziation). <i>Das UML-Klassendiagramm mit den Beziehungen befindet sich hinter dieser Tabelle.</i>	4
2.1	Die Methode <i>neueStadt(...)</i> ist eine Prozedur und wird mit dem Parameter <i>Name</i> vom Datentyp <i>String</i> aufgerufen. Mit Hilfe der while-Schleife wird ein freier Platz im Feld <i>Staedte</i> gesucht. Wenn ein freier Platz gefunden wird, wird eine neue Stadt mit dem als Parameter übergebenen Namen angelegt und dort gespeichert, anderenfalls wird eine Fehlermeldung ausgegeben.	4
2.2	Die Methode <i>loescheStadt(...)</i> ist eine Prozedur und wird mit dem Parameter <i>Name</i> vom Datentyp <i>String</i> aufgerufen. In der for-Schleife wird mit Hilfe zweier if-Anweisungen für jede existierende Stadt geprüft, ob eine Direktverbindung zu dieser Stadt vorhanden ist. Wenn dies der Fall ist, wird die Direktverbindung gelöscht. Damit werden alle Direktverbindungen, in denen die zu löschende Stadt vorkommt, gelöscht. Zum Schluss wird in der Variablen <i>i</i> der Feldindex der zu löschenden Stadt gespeichert. Mit <i>Staedte[i] := nil</i> wird die Stadt aus dem Fernbusnetz entfernt.	6

Aufg.	erwartete Leistungen	BE
3	<pre> procedure TFernbusnetz.neueDirektverbindung(vonStadt, nachStadt: string; Entfernung: integer); var vonNr, nachNr: integer; ersteVerbindung, neueVerbindung: TDirektverbindung; begin if not existiertDirektverbindung(vonStadt, nachStadt) then begin vonNr:= gibFeldIndex(vonStadt); nachNr:= gibFeldIndex(nachStadt); ersteVerbindung:= Staedte[vonNr].getErsteDirektverbindung(); neueVerbindung:= TDirektverbindung.create(nachNr, Entfernung); Staedte[vonNr].setErsteDirektverbindung(neueVerbindung); neueVerbindung.setNaechsteVerbindung(ersteVerbindung); end; end; </pre>	6
4.1	Änderung von Zeile 01: <pre> procedure TFernbusnetz.suche(vonNr, nachNr: integer; Pfad: string; Laenge: integer); </pre> Änderung von Zeile 08: <pre> writeln (' Pfad gefundenen: ', Pfad, ' Länge: ', Laenge); </pre> Änderung von Zeile 15: <pre> suche(neueNr, nachNr, neuerPfad, Laenge+aktVerb.getEntfernung); </pre> Änderung von Zeile 29: <pre> suche(vonNr, nachNr, Staedte[vonNr].getName(), 0); </pre>	5
4.2	Die Klasse <i>TFernbusnetz</i> erhält die beiden neuen Attribute <i>kuerzesterPfad</i> vom Datentyp <i>String</i> und <i>kuerzesteEntfernung</i> vom Datentyp <i>integer</i> . Sie werden in der Methode <i>sucheVerbindungen(...)</i> mit einem Leerstring und einer sehr großen Zahl initialisiert. Wird bei einem Aufruf von <i>suche(...)</i> ein Weg mit einer kürzeren Entfernung gefunden, dann werden <i>kuerzesterPfad</i> und <i>kuerzesteEntfernung</i> durch die gefundenen Daten ersetzt (Minimumsuche). Nachdem alle möglichen Wege bestimmt wurden, ist der kürzeste Weg im Attribut <i>kuerzesterPfad</i> gespeichert.	5
Summe		35

Das Klassendiagramm zu Aufgabe 1.2



III Bewertung und Beurteilung

Die Bewertung und Beurteilung erfolgt unter Beachtung der nachfolgenden Vorgaben nach § 33 der Oberstufen- und Abiturverordnung (OAVO) vom 20. Juli 2009 (ABl. S. 408), zuletzt geändert durch Verordnung vom 1. August 2017 (ABl. S. 672). Nach § 52 (Übergangsregelungen) ist Folgendes zu beachten:

- Bei der Bewertung und Beurteilung der sprachlichen Richtigkeit in der deutschen Sprache sind die Bestimmungen des § 9 Abs. 12 OAVO in Verbindung mit Anlage 9b in der seit 16. September 2017 geltenden Fassung anzuwenden.
- In den modernen Fremdsprachen sowie den alten Sprachen gelten die Bestimmungen des § 9 Abs. 13 OAVO in Verbindung mit den Anlagen 9b und c bzw. 9d der Verordnung in der bis zum 15. August 2016 geltenden Fassung.

Bei der Berechnung von Prozentwerten und Fehlerindizes gemäß Anlage 9 OAVO werden die berechneten Werte nicht gerundet. Für die Umrechnung von Prozentanteilen der erbrachten Leistungen in Punkte ist Anlage 9a zu § 9 Abs. 12 OAVO anzuwenden. 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 (objektorientierte Modellierung), Q2 (Datenbanken) und Q3 (Konzepte und Anwendungen der theoretischen Informatik) 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 insgesamt 46% der zu vergebenden BE erreicht werden. Ein Prüfungsergebnis von **11 Punkten (gut)** setzt voraus, dass insgesamt 76% 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	
1.1	5			5
1.2	2	2		4
2.1	3	1		4
2.2		6		6
3		3	3	6
4.1		3	2	5
4.2		3	2	5
Summe	10	18	7	35

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