

## **Musteraufgaben für die Abschlussprüfung im Ausbildungsberuf Mathematisch-technische Softwareentwickler**

### Prüfungsbereich "Softwareentwurf und Programmierung"

#### **Vorbemerkung:**

Der Aufgabensatz besteht aus drei Aufgaben, deren Punkteverteilung jeweils vom Aufgabenersteller festgelegt wird.

Der vorliegende Aufgabensatz besteht aus drei Aufgaben. Dabei bildet die erste Aufgabe (Programmieraufgabe) den Schwerpunkt. Die beiden anderen Aufgaben werden aus verschiedenen Bereichen der allgemeinen Informatik gewählt, hier aus den Bereichen Algorithmen und Datenbanken.

Der Quellcode zur Programmierung wird ausschließlich handschriftlich auf Papier entwickelt. Hilfsmittel sind hier\* nicht zugelassen.

#### **1. Aufgabe (60 Punkte) [Korrekte Ergänzung von lückenhaften ISBN-Nummern]**

Im Jahre 1965 entschloss sich W. H. Smith, der damals größte Buchverlag in Großbritannien, seinen gesamten Buchbestand in elektronischer Form zugreifbar zu machen. Zu diesem Zweck entwickelte er in Zusammenarbeit mit verschiedenen Experten des Buchhandels ein einheitliches Buchnummerierungssystem. So entstand 1966 das Standard Book Numbering (SBN) System, das bereits im darauf folgenden Jahr in Betrieb genommen wurde.

Von der Idee überzeugt, machten sich daraufhin Delegationen des Internationalen Komitees für Standardisierung (ISO) daran, das System für eine internationale Verwendung anzupassen. Nach ein paar Treffen von Vertretern der anfänglich am Projekt beteiligten Ländern entstand schließlich das International Standard Book Number (ISBN) System. Dieses System entwickelte sich in der Folge zum weltweit gebräuchlichsten Buchkatalogisierungssystem. Es kommt mittlerweile in fast 150 Ländern zur Anwendung.

Eine ISBN-Nummer ist stets zehnstellig, wobei die Anzahl der Stellen der drei ersten Teile variieren kann. Sie setzt sich immer aus vier Teilen zusammen:

- Gruppenzahl (kann bis zu fünf Stellen enthalten)
- Verlagsnummer (kann bis zu sieben Stellen enthalten)
- Titelnnummer (kann bis zu sechs Stellen enthalten)
- Prüfziffer (enthält immer genau eine Stelle)

Diese vier Teile werden durch Bindestriche (oder seltener Leerzeichen) abgetrennt.

Die Prüfziffer erlaubt, die Gültigkeit einer ISBN-Nummer festzustellen. Sie wird so gewählt, dass die ganze Nummer folgende mathematische Eigenschaft erfüllt:

Man multipliziert die erste Stelle mit 10, die zweite Stelle mit 9, die dritte mit 8, und so weiter bis zur neunten Stelle (mal 2), und addiert alle erhaltenen Produkte. Wenn man zu dieser Zahl die Prüfziffer dazuaddiert, so muss ein Vielfaches von 11 entstehen. Als Besonderheit kann die Prüfziffer auch den Buchstaben X annehmen, der dann als Zahlenwert 10 interpretiert wird.

---

\* In der Regel sind zur Bearbeitung der Aufgaben keine Hilfsmittel zugelassen. Abweichungen davon werden mit der Einladung zur Prüfung mitgeteilt.

Die Anwendung von Robert Sedgewicks "Algorithmen" auf die ISBN-Nummer 3-8931-9301-4 der Addison-Wesley-Ausgabe ergibt die folgende Rechnung

$$3 \cdot 10 + 8 \cdot 9 + 9 \cdot 8 + 3 \cdot 7 + 1 \cdot 6 + 9 \cdot 5 + 3 \cdot 4 + 0 \cdot 3 + 1 \cdot 2 + \text{Prüfziffer } 4 = 264.$$

Diese Zahl ist durch 11 teilbar  $264 = 24 \cdot 11$ , also ist es eine gültige ISBN-Nummer.

...

- a) Entwerfen Sie eine Klasse zur Speicherung von ISBN-Nummern und darin eine Methode, die ISBN-Nummern restauriert, in denen genau eine Stelle verloren gegangen ist. Dieser Methode wird als Parameter ein defekter ISBN-Code als String übergeben. Sie liefert als Ergebnis die korrigierte ISBN-Nummer zurück.

Übergabeparameter:

Die Eingabedatei enthält genau eine ISBN-Nummer, in der eine beliebige Stelle durch einen Punkt ersetzt worden ist.

Beispiel für eine defekte ISBN-Nummer: **3-48.-06941-1**

Ausgabeformat:

Ihr Programm soll die ISBN-Nummer ausgeben, die zur Eingabe passt.

Beispielausgabe: **3-486-06941-1**

- b) In einem String-Array der Länge 10 befinden sich korrigierbar lückenhafte ISBN-Nummern. Entwickeln Sie eine Testklasse die diese ISBN-Nummern korrigiert und auf dem Bildschirm ausgibt.

Insgesamt sind für die Teilaufgaben a) und b) abzugeben

- ein Klassenentwurf mit einer Beschreibung der erforderlichen Datenstrukturen in UML-Darstellung.
- der Programmcode in einer der zugelassenen Programmiersprachen mit sinnvoller, knapper Inlinedokumentation unter Angabe der verwendeten Programmiersprache (und des einzusetzenden Compilers).

## **2. Aufgabe (20 Punkte) [Sortialgorithmen]**

In einer Bank sollen Buchungsdaten sortiert werden. Die auflaufenden Buchungsdaten müssen dazu regelmäßig neu sortiert werden.

Die Entwicklungsabteilung plant, den Algorithmus „*HeapSort*“ zu nutzen.

- a) Nennen Sie einen Vorteil des Algorithmus *HeapSort* gegenüber dem Algorithmus *BubbleSort*.
- b) Simulieren Sie den *HeapSort* unter Benutzung der folgenden Zahlenfolge.

3	7	4	9	11	5	7
---	---	---	---	----	---	---

- c) Geben Sie für den *HeapSort* eine Aufwandsabschätzung an (worst case, average case, best case).

...

**3. Aufgabe (20 Punkte) [Datenbanken]**

Ein Vermietungsunternehmen für Baumaschinen will in Zukunft zur Unterstützung ihrer Rechnungserstellung ein Datenbanksystem nutzen.

Ihre Aufgabe besteht darin, einen Teil dieser Anwendung zu erstellen.

Als Grundlage für ihre Arbeit haben Sie tabellarisch folgende Daten gegeben:

Kundenname	Taunus AG	Taunus AG
Ort	Bad Homburg	Bad Homburg
Gemietete Maschinenart	LKW	Bagger
Typ der Maschine	VO12B	PL20
KFZ-Kennzeichen	F - XX 123	MZ - B 56
Maschinennummer	1500	1600
Hersteller	VEVA	LAIN
Mietzeitraum	10.08.2001 - 23.08.2001	1.9.2001 - 5.9.2001
Mietpreis pro Tag	195,- EUR	395,- EUR
Mietvertrag abgeschlossen bei	Niederlassung Süd Mannheim Ansprechpartner: Frau Schneider	Niederlassung West Mainz Ansprechpartner: Herr Müller
Kundennummer	1000	1000
Gefahrene Kilometer	1445	84
Preis je km	2,55 EUR	6,50 EUR
Befähigungsnachweis zum Führen der Maschine	2	S

Folgende zusätzliche Bedingungen gelten:

- Jeder Maschinentyp hat nur einen Hersteller
- Pro Maschinentyp sind der Preis je Kilometer und der Mietpreis pro Tag fest
- Alle Maschinen können bei allen Niederlassungen gebucht werden.
- Die erforderlichen Befähigungsnachweise berechtigen zum Führen eines bestimmten Maschinentyps.

a) Leiten Sie aus diesen Daten Tabellen für ein relationales Datenbanksystem ab. Die Tabellen sollen der dritten Normalform genügen.

b) Tragen Sie die entsprechenden Ausprägungen in die Relationen.  
Die Fremdschlüssel sind mit einem „F\_“ vor ihrer Bezeichnung zu kennzeichnen.  
(Beispiel: F\_KursNr)

**Lösungs- und Bewertungshinweise****1. Aufgabe (60 Punkte) [Korrekte Ergänzung von lückenhaften ISBN-Nummern]**

a) [50 P.]

UML-Darstellung 15 P.

Programmierung (45 P.)

- Datenstrukturen 15 P.
- Analyse der Länge der Teile 10 P.
- Finden und korrigieren der defekten Stelle (20 P.)
  - Normalfälle 15 P.
  - "X" 3 P.
  - Prüfziffer ist defekt 2 P.

b) [10 P.]

UML-Darstellung 3 P.

Umsetzung in Programmcode 7 P.

**2. Aufgabe (20 Punkte) [Sortieralgorithmen]**

a)(4 Punkte) Heapsort ist zwar nicht rekursiv aber dafür sehr stabil was die Anzahl der Vergleiche und Vertauschungen angeht. Das Verfahren hat immer eine asymptotische Laufzeit. Bubblesort dagegen hat in der Regel quadratischen Aufwand. (Ohne Optimierung)

b)(10 Punkte) Die Simulation des Algorithmus der Zahlenfolge gliedert sich in zwei Teile.

1. Bildung des Heap aus der Zahlenfolge
2. Auslesen der Daten aus dem Heap sowie der Neuorganisation des Heaps

Somit ergibt sich die folgende Simulation

Bemerkung / Index	1	2	3	4	5	6	7
Ausgangsdatensatz	3	7	4	9	11	5	7
1. Teil							
Nach der Heap Bildung	11	9	7	3	7	5	4
2. Teil							
1. Durchgang mit reheap	9	7	7	3	4	5	11
2. Durchgang mit reheap	7	5	7	3	4	9	11
3. Durchgang mit reheap	7	5	4	3	7	9	11
4. Durchgang mit reheap	5	3	4	7	7	9	11
5. Durchgang mit reheap	4	3	5	7	7	9	11
6. Durchgang	3	4	5	7	7	9	11

Ein 7. Durchgang ist nicht erforderlich, da die einelementige Restliste immer sortiert ist.

...

c) (6 Punkte) Die Laufzeit für verschiedene Eingabefolgen variiert nur geringfügig. Daraus ergibt sich:

$$C_{\min}(N) = C_{\text{average}}(N) = C_{\max}(N) = O(N \cdot \log N)$$

für die Anzahl der Vergleiche und

$$M_{\min}(N) = M_{\text{average}}(N) = M_{\max}(N) = O(N \cdot \log N)$$

für die Anzahl der Vertauschungen.

### 3. Aufgabe (20 Punkte) [Datenbanken]

T_Kunden	<u>KundenNr</u> Kundenname Ort
T_Niederlassungen	<u>NiederlassungsNr</u> Niederlassung Ort Ansprechpartner
T_Maschinenarten	<u>Typ</u> Maschinenart Führungsnachweis Herstellername Preis_je_km Mietpreis_pro_Tag
T_Maschinen	<u>MaschinenNr</u> F_Typ KFZKennzeichen
T_Mieten	<u>F_Kundennummer</u> <u>F_Maschinennummer</u> F_Niederlassungsnummer Miete_von Miete_bis Gefahren_km

- Je Relation in 3. NF 5 Punkte
- Jeweils 1 Punkt Abzug für
  - fehlende Fremdschlüssel bzw. fehlende Markierung als Fremdschlüssel
  - fehlende Primärschlüssel
  - nicht in 3. NF
  - falsche Beziehungen
  - nicht Beachtung des Geschäftsmodells.