



Dr. Stephan Mehl

Fachinformatiker/-in

Fachrichtung Anwendungsentwicklung

Prüfungstrainer Abschlussprüfung
Prüfungsteil B · Ganzheitliche Aufgaben I
– Fachqualifikationen –
Übungsaufgaben und erläuterte Lösungen

Bestell-Nr. 756

Ihre Meinung ist uns wichtig!

Du hast Fragen, Anregungen oder Kritik zu diesem Produkt?

Das U-Form Team steht dir gerne Rede und Antwort.

Direkt auf

www.facebook.com/pruefungcheck

fragen, diskutieren, stöbern und weiteres Wichtige und
Wissenswerte rund um Ausbildung erfahren

oder einfach eine kurze E-Mail an

feedback@u-form.de

WIR HABEN NOCH MEHR!

In unserem Online-Shop findest du das komplette Angebot für eine optimale
Prüfungsvorbereitung Fachinformatikerin / Fachinformatiker.

JETZT ENTDECKEN



Titelbild

Julien Eichinger – Fotolia.com

COPYRIGHT

U-Form Verlag · Hermann Ullrich GmbH & Co. KG

Cronenberger Straße 58 · 42651 Solingen

Telefon 0212 22207-0 · Telefax 0212 22207-63

Internet: www.u-form.de · E-Mail: uform@u-form.de

Alle Rechte liegen beim Verlag bzw. sind der Verwertungsgesellschaft Wort, Untere Weidenstr. 5, 81543 München, Telefon 089 514120, zur treuhänderischen Wahrnehmung überlassen. Damit ist jegliche Verbreitung und Vervielfältigung dieses Werkes – durch welches Medium auch immer – untersagt.

8. Auflage 2021 · ISBN 978-3-95532-756-9

Die IHK-Abschlussprüfungsklausur speziell für Fachinformatiker/-innen Anwendungsentwicklung, die Ganzheitliche Aufgabe I, ist recht einheitlich strukturiert. Es werden fünf Aufgaben (die sogenannten Handlungsschritte) gestellt, die die folgenden Themen abdecken:

1. Projektmanagement, UML
2. Entwicklung eines Algorithmus
3. Objektorientierte Programmierung
4. Datenbankentwurf
5. SQL

Von diesen fünf Handlungsschritten sind vier auszuwählen. Zu Beginn der Prüfung steht man also vor dem Problem, welche Aufgabe man streichen soll. Für viele Auszubildende hängt die Antwort davon ab, in welchem Bereich (Programmierung oder Datenbanken) ihr persönlicher Ausbildungsschwerpunkt gelegen hat. Davon abgesehen gibt es aber Erfahrungswerte zu den einzelnen Themenbereichen, die im Folgenden kurz beschrieben werden sollen.

Die erste Aufgabe – Projektmanagement – fällt erfahrungsgemäß stets besonders gut aus; sozusagen eine Aufgabe zum Anwärmen. Falls UML-Diagramme gefragt sind, werden meist in einem separaten Belegsatz Informationen zu diesem Diagrammtyp beigelegt. Diese Hinweise sollten unbedingt ernst genommen werden, denn manche Aufgaben verlangen selten genutzte Konstruktionen, die in diesen Datenblättern erklärt werden.

In manchen Prüfungen besteht diese Aufgabe eher aus sprachlich formulierten Erklärungen als aus formalen technischen Darstellungen und wird daher von manchen Prüflingen gemieden. Sie besteht aber oft aus vielen Unterpunkten, so dass es weniger stark ins Gewicht fällt, wenn man einen einzelnen Unterpunkt falsch beantwortet. Keinesfalls sollte man Antwortfelder leer lassen – was nicht da steht, kann nicht richtig sein.

Aufgabe 2 verlangt die Darstellung eines Algorithmus und benötigt daher zunächst eine längere Planungsphase und anschließend viel Zeit für die Reinschrift. Man kann diesen höheren Zeitbedarf zwar bei den anderen Aufgaben ausgleichen, trotzdem wird diese Aufgabe zu Recht von den meisten Prüflingen gestrichen. Hier ist es besonders wichtig, die Aufgabenstellung genau zu lesen, um keinen falschen Ansatz zu wählen. Häufig werden Funktionen vorgegeben, die dann auch unbedingt verwendet werden sollten.

Die Komplexität dieser Aufgabe variiert aber von Prüfung zu Prüfung stark. Man sollte sie daher nicht vorschnell außen vor lassen, bevor man nicht die anderen Aufgaben gesehen hat.

Es lohnt sich also, diesen Aufgabenbereich besonders zu üben. In den Kapiteln dieses Prüfungstrainers wird daher Wert darauf gelegt, stets eine oder mehrere solcher Programmieraufgaben anzubieten, bei denen der zu erstellende Pseudocode bzw. das Struktogramm keine übermäßige Länge hat, wohl aber die üblicherweise erwartete Komplexität (z. B. verschachtelte Bedingungen oder Wiederholungsstrukturen und vor allem die Arbeit mit klassischen Arrays).

Der dritte Handlungsschritt behandelt den Bereich der objektorientierten Programmierung, meist mit Angabe von Klassendiagrammen, die ergänzt oder kommentiert werden müssen oder zu denen Pseudocode zu erstellen ist. Eventuell werden diese Klassendiagramme in zwei Ausschnitten präsentiert, wobei der erste Ausschnitt Teil des zweiten ist.

Hervorzuheben ist, dass seit 2018 dieses Klassendiagramm einem Entwurfsmuster (Design Pattern) folgt. Die Kenntnis dieses Entwurfsmusters ist nicht erforderlich, das Diagramm wird dadurch aber komplexer als früher. Entsprechende Übungsaufgaben finden sich in der vorliegenden Neuauflage dieses Prüfungstrainers in den Kapiteln 2, 3, 6 und 10. Die übrigen Kapitel enthalten Aufgaben zu den Grundbegriffen der objektorientierten Programmierung, die nach wie vor auch in der IHK-Klausur den Löwenanteil der Punkte für die 3. Aufgabe liefern.

An dieser Stelle geht ein herzliches Dankeschön an meinen Kollegen Robert Wittek für seine Beratung hinsichtlich der Entwurfsmuster. Eventuelle Fehler gehen natürlich auf meine Kappe.

Im vierten Handlungsschritt enthalten die meisten Prüfungen eine relativ einfache Aufgabe zum Datenbankentwurf, auch als Ausgleich zu der zeitaufwändigen Programmieraufgabe. Allerdings gibt es auch hier zwei Fallstricke. Zum einen ist zu beachten, ob ein Entity-Relationship-Modell oder ein Datenbankschema erstellt werden soll. Wird bei der Lösung das falsche Modell abgeliefert, dürfte es Punktabzug geben. Zum zweiten soll das geforderte Datenbankschema in der Regel normalisiert sein. Dazu muss die Aufgabenstellung genau gelesen werden, denn oftmals enthält sie Hinweise auf spezielle Abhängigkeiten, die in ähnlichen Aufgaben nicht vorkommen.

Der fünfte Handlungsschritt erwartet die Formulierung von SQL-Anweisungen. Dazu muss zunächst eine vorgegebene Tabellenstruktur genau analysiert werden. Hier variieren die Prüfungen stark; manchmal enthalten sie mehrere, eher einfache Aufgaben, gelegentlich aber auch wenige, sehr komplexe.

Aus all dem folgt, dass es keine allgemeine Empfehlung geben kann, diesen oder jenen Handlungsschritt unbedingt zu streichen. Nehmen Sie sich zu Beginn der Prüfung Zeit, alle Aufgaben durchzusehen und sich dann zu entscheiden. Vor allem sollten Sie nicht zu viel Zeit mit einer einzigen Aufgabe, z. B. dem Algorithmus, verbringen. Das Ziel ist, aus allen vier bearbeiteten Aufgaben möglichst viele Teilpunkte zu erreichen. Notfalls skizzieren Sie z. B. bei der Algorithmus-Aufgabe die grobe Struktur, ohne die Details auszuarbeiten – das ist besser, als ein leeres Blatt abzugeben.

Die beschriebene Abfolge von Aufgaben ist ein Erfahrungswert der letzten Jahre, keine zwingende Vorschrift. Einige Kapitel dieses Prüfungstrainers behandeln die fünf Themen daher in einer abgewandelten Reihenfolge.

Aufgabenstellungen sind Interpretationssache. Deswegen enthalten die Musterlösungen der Originalprüfungen häufig den Hinweis, dass auch alternative Lösungen möglich seien. Das heißt jedoch nicht, dass es erlaubt ist, Aufgaben beliebig misszuverstehen, insbesondere nicht so, dass sie deutlich einfacher werden. Auch diese Funktion möchte der vorliegende Prüfungstrainer mit den angebotenen Musterlösungen erfüllen: eine Übersicht zu geben, welche Antworten auf welche Fragen üblicherweise erwartet werden. Sollte eine Aufgabenstellung trotz sorgfältiger Formulierung allzu missverständlich sein, freuen sich Autor und Verlag über entsprechende Hinweise. Trotzdem wird ein „Restrisiko“ stets bleiben. In der Prüfung selbst empfiehlt es sich, im Zweifelsfall einen Hinweis in den Lösungsbogen zu schreiben, wie man die Aufgabe verstanden hat.

Wir wünschen allen Leserinnen und Lesern viel Spaß mit der vorliegenden Aufgabensammlung – möge sie einen Beitrag zu noch besseren Prüfungsergebnissen leisten!

	Seite
Vorwort	3 – 4
Arbeitsanleitung	7 – 10
1. Prüfung: Die Stadtbibliothek	
Ausgangssituation	13
1. Handlungsschritt – 5. Handlungsschritt.....	13 – 16
2. Prüfung: Essen auf Rädern	
Ausgangssituation	17
1. Handlungsschritt – 5. Handlungsschritt.....	17 – 22
3. Prüfung: Konzertkarten online	
Ausgangssituation	23
1. Handlungsschritt – 5. Handlungsschritt.....	23 – 27
4. Prüfung: Der Parkscheinautomat	
Ausgangssituation	29
1. Handlungsschritt – 5. Handlungsschritt.....	29 – 32
5. Prüfung: Der Autoverleih	
Ausgangssituation	33
1. Handlungsschritt – 5. Handlungsschritt.....	33 – 37
6. Prüfung: Das Bauamt	
Ausgangssituation	39
1. Handlungsschritt – 5. Handlungsschritt.....	39 – 42
7. Prüfung: Die Ferienwohnungsvermittlung	
Ausgangssituation	43
1. Handlungsschritt – 5. Handlungsschritt.....	43 – 47
8. Prüfung: Der Multiple Choice-Test	
Ausgangssituation	49
1. Handlungsschritt – 5. Handlungsschritt.....	49 – 52
9. Prüfung: Die Dokumentenverwaltung	
Ausgangssituation	53
1. Handlungsschritt – 5. Handlungsschritt.....	53 – 57
10. Prüfung: Der Güterbahnhof	
Ausgangssituation	59
1. Handlungsschritt – 5. Handlungsschritt.....	59 – 66

Inhaltsverzeichnis Lösungsteil

	Seite
Vorwort Lösungsteil.....	69
Lösungen zur 1. Prüfung: Die Stadtbibliothek	
1. Handlungsschritt – 5. Handlungsschritt	71 – 76
Lösungen zur 2. Prüfung: Essen auf Rädern	
1. Handlungsschritt – 5. Handlungsschritt	77 – 80
Lösungen zur 3. Prüfung: Konzertkarten online	
1. Handlungsschritt – 5. Handlungsschritt	81 – 85
Lösungen zur 4. Prüfung: Der Parkscheinautomat	
1. Handlungsschritt – 5. Handlungsschritt	87 – 90
Lösungen zur 5. Prüfung: Der Autoverleih	
1. Handlungsschritt – 5. Handlungsschritt	91 – 95
Lösungen zur 6. Prüfung: Das Bauamt	
1. Handlungsschritt – 5. Handlungsschritt	97 – 101
Lösungen zur 7. Prüfung: Die Ferienwohnungsvermittlung	
1. Handlungsschritt – 5. Handlungsschritt	103 – 106
Lösungen zur 8. Prüfung: Der Multiple Choice-Test	
1. Handlungsschritt – 5. Handlungsschritt	107 – 110
Lösungen zur 9. Prüfung: Die Dokumentenverwaltung	
1. Handlungsschritt – 5. Handlungsschritt	111 – 115
Lösungen zur 10. Prüfung: Der Güterbahnhof	
1. Handlungsschritt – 5. Handlungsschritt	117 – 122



Aufgabenteil

Ausgangssituation

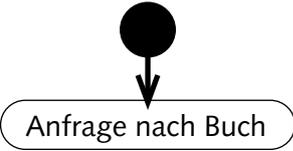
Die Stadtbibliothek Buchholz möchte ihr veraltetes Katalog- und Ausleihsystem modernisieren. Sie erhalten den Auftrag, die folgenden Arbeitsschritte auszuführen:

1. ein Aktivitätsdiagramm für die Organisation der Buchbeschaffung erstellen
2. den Code für eine Hashfunktion zur schnelleren Suche von Autorennamen erstellen und evaluieren
3. ein Objektdiagramm sowie ein Struktogramm für die Suche in einem Binärbaum entwickeln
4. eine Datenbank in die 3. Normalform überführen
5. SQL-Abfragen festlegen

1. Handlungsschritt (25 Punkte)

Die Stadtbibliothek Buchholz verspricht ihren Leserinnen und Lesern, jedes gewünschte Buch zu beschaffen. Sofern der Bibliothekar das Buch nicht im eigenen Bestand findet, bittet er die Beschaffungsstelle, das allgemeine Interesse an diesem Buch in Relation zu seinem Beschaffungspreis zu beurteilen. Weniger interessante oder zu teure Bücher werden nach Möglichkeit über die Fernleihstelle besorgt. Interessante Bücher oder solche, die nicht per Fernleihe erhältlich sind, werden angeschafft. Da die Fernleihstelle einige Zeit für die Ermittlung der Lieferbarkeit benötigt, sucht die Beschaffungsstelle in dieser Zeit sicherheitshalber bereits nach einem Buchsponsor. Notfalls wird das Buch aber auch aus Eigenmitteln beschafft.

Zeichnen Sie hierzu ein Aktivitätsdiagramm.

Leser	Bibliothekar	Beschaffungsstelle	Fernleihstelle
			

2. Handlungsschritt (25 Punkte)

Häufig fragen Leser nach Büchern eines bestimmten Verfassers. Da die Stadtbibliothek einen Buchbestand von ca. 300.000 Büchern, verfasst von ca. 180.000 Autoren hat, sollen die Namen dieser Autoren so abgespeichert werden, dass sie möglichst schnell gefunden werden.

Ihr Projektleiter schlägt vor, die Autorennamen mit Hilfe einer Hashfunktion zu verschlüsseln. Diese Funktion bildet einen Namen auf eine Zahl zwischen 0 und 280000 ab, die als Arrayindex verwendet wird. Unter diesem Index findet sich dann der Autorennamen. In einer späteren Ausbaustufe soll hier ein Verweis auf die Liste aller Bücher von Autoren dieses Namens stehen.

1. Prüfung – Die Stadtbibliothek

Zur Ermittlung des Hashwertes wird jeder Buchstabe des Namens zunächst in einen Kleinbuchstaben und dann in dessen ASCII-Wert minus 96 konvertiert. Ein 'A' oder 'a' zählt daher 1, ein 'Z' oder 'z' 26. Alle Zahlenwerte werden addiert und mit 100 sowie der Länge des Namens multipliziert. Das Ergebnis wird modulo 280000 berechnet.

Einige Beispiele:

Name	Hashwert
Bach	5600
Goethe	36000
Klitt	36000
Schiller	68800
Shakespeare	118800
Aristoteles	157300
Dürrenmatt	269000
Droste-Hülshoff	110000 (=390000 % 280000)

Sollte die errechnete Arrayposition schon von einem anderen Autor belegt sein, wird die jeweils nächste freie Arrayposition genommen.

- Zeichnen Sie ein Struktogramm für eine Suchfunktion, die einen Autorennamen als Parameter übernimmt und die Arrayposition (falls vorhanden) bestimmt, an der dieser Name tatsächlich abgespeichert ist. Ist der Name nicht vorhanden, soll der Wert -1 zurückgegeben werden. Die Funktion soll die oben angegebene Hashcodierung verwenden und ggf. die nachfolgenden Arraypositionen überprüfen. (14 Punkte)
- Nennen Sie zwei Gründe dafür, warum das Array für ungefähr 180000 Autoren die Dimension 280000 statt 180000 hat. (2 Punkte)
- Die Länge eines Namens hat einen beträchtlichen Einfluss auf den mit dem angegebenen Verfahren errechneten Hashwert. Welcher Nachteil ergibt sich hieraus? (3 Punkte)
- Ein Kollege fragt Sie, ob die Arbeit mit einer Hashfunktion wirklich so viel günstiger sei als eine Suche in einem unsortierten oder einem klassisch alphabetisch sortierten Array. Füllen Sie als Antwort die folgende Tabelle aus, indem Sie in jeder Zeile das durchschnittlich schnellste Verfahren mit einer 1, das zweitschnellste mit einer 2 und das langsamste mit einer 3 kennzeichnen. (6 Punkte)

	unsortiertes Array	alphabetisch sortiertes Array	nach Hashwerten organisiertes Array
Suche nach einem vorhandenen Element			
Suche nach einem nicht vorhandenen Element			
Einfügen eines neuen Elements			

3. Handlungsschritt (25 Punkte)

Ebenso häufig wie nach Autorennamen fragen die Leser nach Buchtiteln. Hier möchte Ihr Projektleiter statt der Hashsuche eine Suche in einem binären Baum implementieren. Dazu hat er die folgende Klasse entworfen.

Titel
+Wortlaut: String
+Buchnummer: int
+links: Titel
+rechts: Titel
+suchen(String): Titel

Unter dem Attribut "links" wird ein Verweis auf einen alphabetisch kleineren (davor liegenden) Titel abgespeichert, unter "rechts" ein Verweis auf einen alphabetisch größeren.

- Zeichnen Sie ein Objektdiagramm mit drei Objekten für die Titel "Programmieren für Anfänger" (Buchnummer 123456), "Algorithmen für alle" (Buchnummer 54321) und "Sortieren leicht gemacht" (Buchnummer 33333). (6 Punkte)
- Geben Sie den Pseudocode für die Methode "suchen" an. Diese Methode übernimmt den Wortlaut des zu suchenden Titels als Parameter und sucht ihn je nach alphabetischer Sortierung im linken oder rechten Zweig des Binärbaums. Wird der Titel nicht gefunden, gibt die Methode "null" zurück. (14 Punkte)
- Wovon hängt die Effizienz dieses Suchverfahrens ab? (5 Punkte)

4. Handlungsschritt (25 Punkte)

Die Detailinformationen zu den Büchern wurden bisher in einer Tabelle mit den folgenden Spalten verwaltet:

Inventurnummer
Autor 1 (Name, Vorname)
Autor 2 (Name, Vorname)
Autor 3 (Name, Vorname)
Titel
Verlag
Ort des Verlags
ISBN
Signatur
Schlagwort 1 (Name, Kategorie)
Schlagwort 2 (Name, Kategorie)
Schlagwort 3 (Name, Kategorie)

Besitzt die Bibliothek mehrere Exemplare desselben Buchs, so werden die betreffenden Informationen für jedes Exemplar unter dessen Inventurnummer wiederholt. Alle Exemplare desselben Buchs haben dieselbe Signatur.

Überführen Sie diese Tabelle in ein Datenbankschema in der 3. Normalform.

Kennzeichnen Sie alle Primärschlüssel mit PK und alle Fremdschlüssel mit FK.

Lösungsteil

Die in diesem Band vorgestellten Lösungen sind Musterlösungen. Das bedeutet, dass es in der Regel Alternativen gibt. Wie schon im Vorwort zum Aufgabenteil erwähnt, heißt dies jedoch nicht, dass man sich eine Aufgabenstellung so zurechtbiegen darf, dass die Lösung möglichst einfach ist. Die Komplexität einer Aufgabe, die sich in den hier angegebenen Musterlösungen widerspiegelt, sollte sich auch in alternativen Lösungen wiederfinden. Einige der im Folgenden vorgestellten Lösungen enthalten entsprechende Hinweise.

Zwei regelmäßig wiederkehrende Situationen, in denen andere Lösungen entwickelt werden, sollen kurz thematisiert werden:

SQL:

Viele SQL-Abfragen beziehen mehrere Tabellen ein, die miteinander verknüpft werden müssen. Dies geschieht entweder über eine explizite INNER JOIN-Klausel, über die Verknüpfung der Schlüssel innerhalb der WHERE-Bedingung oder über eine Unterabfrage. Aus Platzgründen wird lediglich die zweite Möglichkeit angegeben.

Pseudocode:

Programmieraufgaben erwarten entweder reinen Pseudocode oder aber Struktogramme oder Programmablaufpläne, die wiederum Pseudocode enthalten. Hier stellen viele Prüflinge die Frage, ob nicht auch Programmcode einer bestimmten Programmiersprache wie C# oder Java angegeben werden könne. Nun stellt die Forderung nach Pseudocode eine Hilfe dar, die die Prüflinge davor bewahren soll, eine bestimmte Programmiersprache zu beherrschen oder Syntaxdetails korrekt wiederzugeben. Darüber hinaus gibt es keine Norm, wie sehr sich Pseudocode von der Syntax irgendeiner Programmiersprache unterscheiden muss. Allerdings bieten viele Programmiersprachen Möglichkeiten an, Aufgabenstellungen mit Hilfe fertiger Funktionen zu lösen, anstatt den geforderten Algorithmus selbst zu programmieren. Beispielsweise können Elemente in Listen mit Hilfe einer Methode „Contains“ gefunden werden, statt selbst eine Schleife mit entsprechenden Tests zu entwickeln. Solche Verkürzungen der Aufgabenstellung sind selbstverständlich nicht zulässig. Insofern ist Pseudocode eben nicht mit C#- oder Java-Code gleichzusetzen.

Bei Aufgaben, die eine Erläuterung von Fachbegriffen oder Arbeitsmethoden verlangen, sind die Musterlösungen in diesem Band bewusst ausführlicher formuliert, als dies einem Prüfling unter Zeitdruck möglich wäre. Dies soll eine Hilfe für diejenigen Leserinnen und Leser darstellen, die mit dem entsprechenden Begriff weniger vertraut sind. Auch im Rahmen einer Prüfung sollte man sich nicht zu kurz fassen - was nicht da steht, kann nicht richtig sein. Für falsche Antworten werden jedoch Punkte abgezogen, auch wenn die falschen durch richtige ergänzt werden.

Alle angegebenen Lösungen sind sorgfältig erstellt worden. Über Hinweise auf eventuell noch vorhandene Fehler sind Autor und Verlag dankbar.

<https://u-form.de/erfolg/3141>

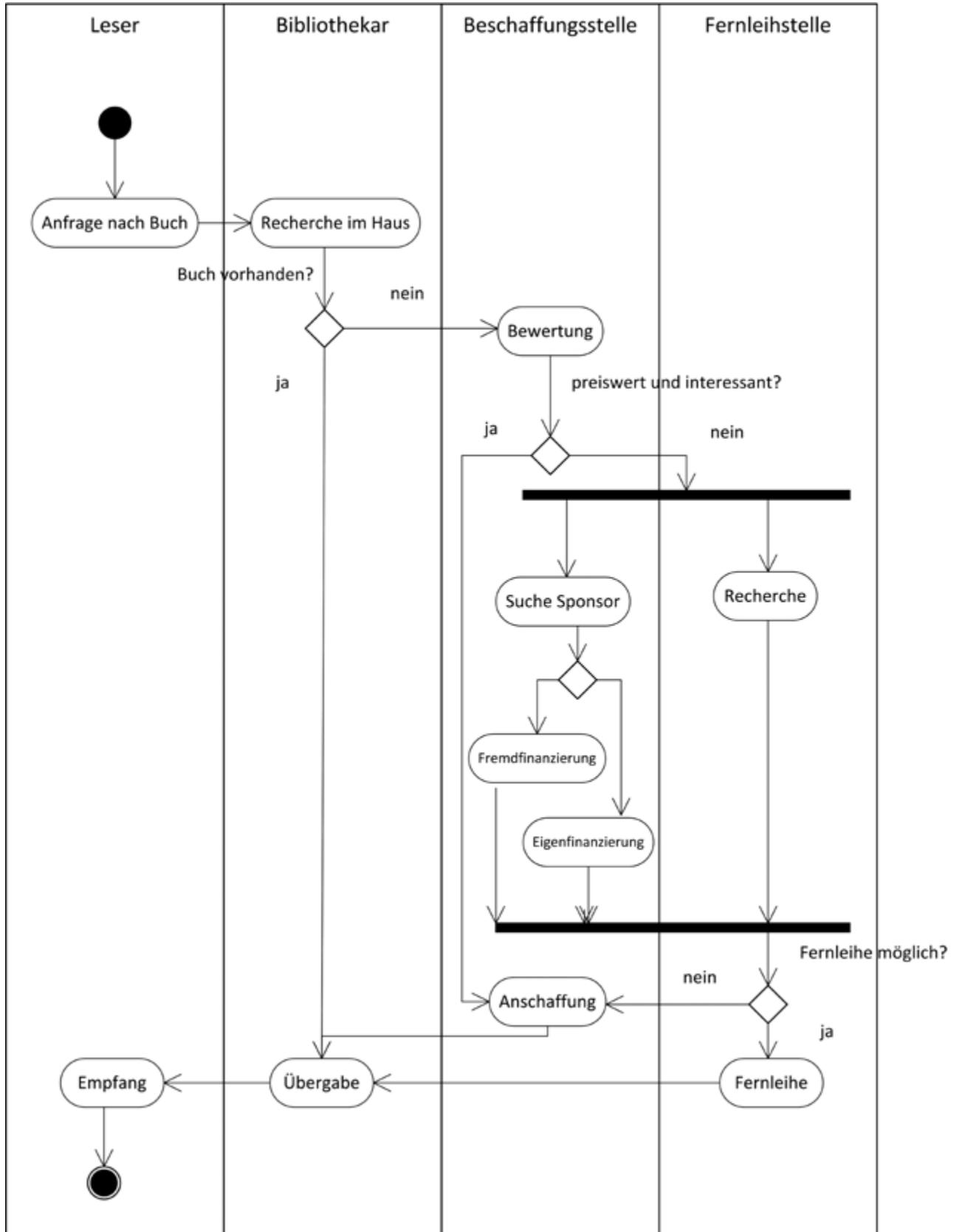
U-FORM ERFOLGSPAKETE

Für deinen krönenden Abschluss: Hol dir jetzt das U-Form Erfolgspaket, welches dich optimal auf alle Prüfungsfächer vorbereitet. Und das Beste daran: Bestehe deine Prüfung oder du erhältst 100% Geld zurück.

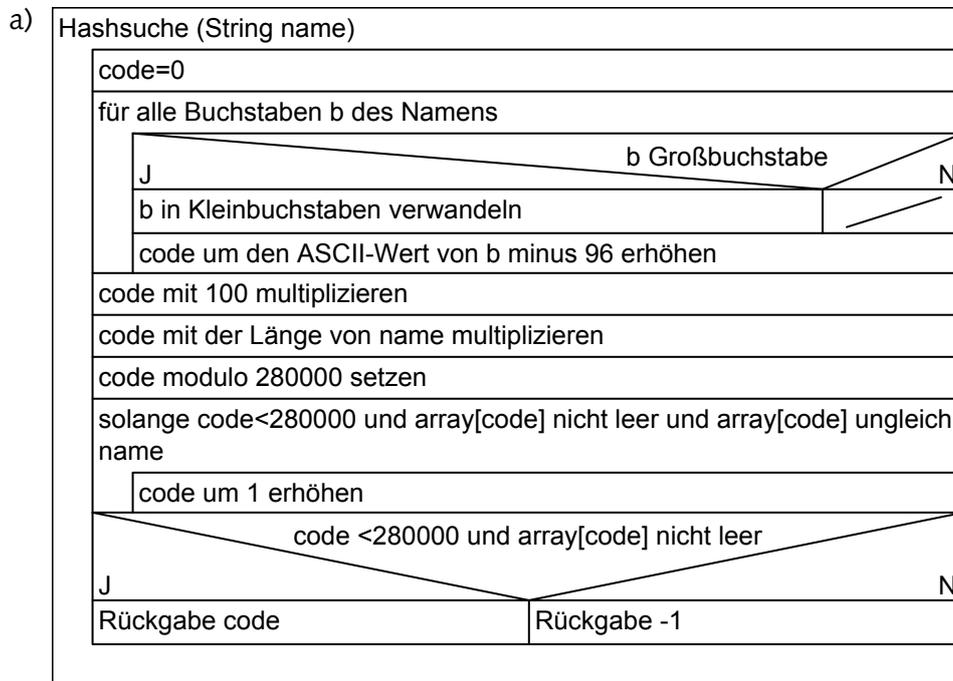
JETZT BESTELLEN



1. Handlungsschritt



2. Handlungsschritt



- b) • Platz für weitere Autoren
- Hashverfahren benötigt freie Positionen, um Kollisionen (mehrere Namen mit gleichem Hashwert) auszugleichen
- c) Die Länge von Namen ist nicht annähernd gleich verteilt. Vielmehr existieren viele Namen mit 5 – 7 Buchstaben und wenige Namen mit weniger als 3 oder mehr als 12 Buchstaben. Dadurch wird es zu einem Engpass ungefähr im Bereich der Indizes 40000 – 60000 kommen.

d)

	unsortiertes Array	alphabetisch sortiertes Array	nach Hashwerten organisiertes Array
Suche nach einem vorhandenen Element	3	2	1
Suche nach einem nicht vorhandenen Element	3	2	1
Einfügen eines neuen Elements	1	3	2

Fortsetzung auf der nächsten Seite

Erläuterung (nicht als Antwort verlangt):

Bei einem unsortierten Array muss bei der Suche nach einem vorhandenen Element im Durchschnitt das halbe Array durchsucht werden, bei einer erfolglosen Suche sogar das ganze. Dafür geht das Einfügen (Anhängen ans Ende des Arrays) sehr schnell.

Bei einem sortierten Array sind erfolgreiche und erfolglose Suche gleich schnell, weil man sie abbrechen kann, wenn man die entsprechende Stelle im Alphabet erreicht hat. Außerdem lässt sich die Suche beschleunigen, indem man nicht am Anfang des Array anfängt, wenn der Anfangsbuchstabe eher am Ende des Alphabets liegt. Dafür müssen zum Einfügen alle nachfolgenden Elemente verschoben werden.

Wie schnell das Hashverfahren ist, hängt von der Anzahl der Kollisionen ab. In einem dicht belegten Bereich kann besonders die erfolglose Suche etwas länger dauern. Im Durchschnitt ist die Hashsuche aber das schnellste der drei Verfahren, weil gezielt an der richtigen Stelle gesucht wird.

3. Handlungsschritt

- a) Hier gibt es drei mögliche Lösungen, je nachdem, welches Objekt als Wurzel des Baums genommen wird. Am effizientesten für die Suche wäre es, den alphabetisch mittleren Titel zu wählen:

<u>Objekt1 : Titel</u>

Wortlaut = Programmieren für Anfänger Buchnummer = 123456 links = Objekt2 rechts = Objekt3

<u>Objekt2 : Titel</u>

Wortlaut = Algorithmen für alle Buchnummer = 54321 links = null rechts = null
--

<u>Objekt3 : Titel</u>

Wortlaut = Sortieren leicht gemacht Buchnummer = 33333 links = null rechts = null
--

Fortsetzung auf der nächsten Seite

Lösungen zur 1. Prüfung – Die Stadtbibliothek

Wählt man "Algorithmen für alle" als Wurzel, ergibt sich:

<u>Objekt1 : Titel</u>
Wortlaut = Algorithmen für alle Buchnummer = 54321 links = null rechts = Objekt2

<u>Objekt2 : Titel</u>
Wortlaut = Programmieren für Anfänger Buchnummer = 123456 links = null rechts = Objekt3

<u>Objekt3 : Titel</u>
Wortlaut = Sortieren leicht gemacht Buchnummer = 33333 links = null rechts = null

Die hier gewählte Anordnung der Objekte symbolisiert die Baumstruktur, ist aber völlig beliebig.

b) suchen (Suchwort: String): Titel

```
falls Suchwort < Wortlaut
  falls links nicht null
    falls Suchwort=links.Wortlaut
      ergebnis = links
    sonst
      ergebnis = links.suchen(Suchwort)
  sonst
    ergebnis = null
sonst
  falls rechts nicht null
    falls Suchwort=rechts.Wortlaut
      ergebnis = rechts
    sonst
      ergebnis = rechts.suchen(Suchwort)
  sonst
    ergebnis = null
Rueckgabe: ergebnis
```

c) Die Effizienz hängt davon ab, wie ausgeglichen (balanciert) der Baum ist. Im Idealfall sind nur die Zeiger auf der untersten Ebene null, sodass der Suchweg von der Wurzel bis zu den Blättern möglichst kurz ist.