Aufgabe A

Ein Zoo bietet eine App an, mit deren Hilfe Tiere bestimmt werden können. Dabei werden bei Verwendung der App für bestimmte Aktionen Punkte an die Nutzenden vergeben. Zur Nutzung der App ist eine vorherige Registrierung nötig. Eine in der App registrierte Person wird als Objekt der Klasse Person verwaltet. In Abbildung 2 im Material ist eine Klassenkarte der Klasse dokumentiert.

1. Erläutern Sie anhand der Klasse Person den grundsätzlichen Aufbau einer Klasse. Gehen Sie dabei auch auf die Sichtbarkeitsattribute (+ und -) ein. [5 BE]

Alle registrierten Personen werden in einer global definierten dynamischen Reihung user vom Typ Person verwaltet.

1. Die Operation juengstesAlter(): Ganzzahl bestimmt, wie alt die jüngsten in der App registrierten Personen sind und gibt dieses Alter als Ganzzahl zurück.  
   Implementieren Sie die Operation in einer im Unterricht verwendeten Programmiersprache. Gehen Sie dabei davon aus, dass die Reihung user nicht leer ist. [4 BE]
2. Erläutern Sie Änderungen an Ihrer Operation, damit diese zusätzlich zum Alter die Anzahl der jüngsten registrierten Personen zurückgibt. [4 BE]

Beim Entwurf rekursiver Algorithmen spielt die Strategie „Teile und herrsche“ eine wichtige Rolle.

1. Erläutern Sie diese Strategie an einem selbst gewählten Beispiel. [4 BE]

In Abbildung 3 ist das Struktogramm der rekursiven Operation sortiereNeu dargestellt, mit deren Hilfe für einen Teilbereich von user eine Sortierung in Bezug auf das Attribut punkte vorgenommen werden kann.

1. Dokumentieren Sie in Abbildung 4 für die gegebene dynamische Reihung user die rekursiven Selbstauf­rufe, die sich bei der Verarbeitung des Aufrufs sortiereNeu(0,2) ergeben. Notieren Sie zu jedem Selbstaufruf ergänzend, welche Belegung die Reihung user zum Zeitpunkt des Selbstaufrufs besitzt. [4 BE]
2. Der rekursive Selbstaufruf sortiereNeu(li, re-k) erfolgt zweimal in der Operation sortiereNeu.   
   Erläutern Sie, dass im Allgemeinen nicht auf den zweiten Aufruf verzichtet werden kann. [4 BE]

Die Tierbestimmungsapp bietet die Möglichkeit der Bestimmung von einigen heimischen Vögeln.

Die zu bestimmenden Vogelbezeichnungen und ihre jeweiligen Merkmale werden in einem global definierten Binärbaum heimischeVoegel vom Typ Zeichenkette gespeichert. Ein solcher Binärbaum wird Entscheidungsbaum genannt. In Abbildung 5 ist beispielhaft ein Entscheidungsbaum dargestellt.

In Abbildung 6 ist die Operation bestimmeVogel gegeben, der als Parameter ein Entscheidungsbaum übergeben wird. Mithilfe der Operation wird der Entscheidungsbaum abhängig von nacheinander vorzunehmenden Nutzereingaben durchlaufen und eine Vogelbezeichnung zurückgegeben.

1. Geben Sie an, welche Werte die Variable merkmal nacheinander angenommen hat, wenn die Operation bestimmeVogel(heimischeVoegel) den Wert "Weißstorch“ zurückgibt. [2 BE]
2. In eine zunächst leere, global definierte dynamische Reihung merkmale vom Typ Zeichenkette sollen alle in heimischeVoegel vorkommenden Merkmale eingetragen werden. Die in den Blättern gespeicherten Vogelbezeichnungen werden nicht in merkmale aufgenommen. Im Baum mehrfach vorkommende Merkmale sollen nur einmalig in merkmale eingetragen werden.

Implementieren Sie in einer im Unterricht verwendeten Programmiersprache eine entsprechende Operation gibMerkmale(b: Binärbaum vom Typ Zeichenkette). [8 BE]

Die App verwendet mehrere verschiedene Entscheidungsbäume, um auch Tiere anderer Gattungen zu bestimmen. Die Tiefe eines Entscheidungsbaums ist definiert als die maximale Anzahl der Merkmale, die zur Bestimmung eines Vogels abgefragt werden. Für den Baum aus Abbildung 5 beträgt die Tiefe beispielsweise 4.

1. Entwickeln Sie ein Struktogramm einer Operation tiefeBestimmen, die einen Entscheidungsbaum als Parameter übergeben bekommt und die zugehörige Tiefe als Ganzzahl zurückgibt. [4 BE]

Im Folgenden sollen für eine natürliche Zahl n die möglichen Tiefen eines Entscheidungsbaums mit n verschiedenen Merkmalen untersucht werden, wobei jedes Merkmal nur genau einmal im Baum vorkommt.

1. Untersuchen Sie, welche Tiefe ein solcher Entscheidungsbaum maximal haben kann und welche Tiefe er mindestens haben muss. [5 BE]

Der Zoo verwaltet Daten in einer Datenbank. Das zugrundeliegende relationale Datenbankschema ist in Abbildung 1 dargestellt.

|  |
| --- |
| Tier(ID, Name, Geburtsjahr, Gattung, Gewicht, ↑GebäudeNr)  Gebäude(GebäudeNr, Name, Größe)  Futtersorte(Nummer, Name, Form, ↑GebäudeNr, ↑LieferantID)  Lieferant(ID, Name, Telefonnummer)  Futter(↑TierID, ↑Futternummer, KGProTag) |

Abbildung 1: Relationales Datenbankschema zum Zoo

1. Untersuchen Sie, ob sich statt des Attributs ID ein anderes einzelnes Attribut der Tabelle Tier als Primärschlüssel für die Tabelle Tier eignet. [3 BE]
2. Entwerfen Sie SQL-Abfragen auf Grundlage des gegebenen Datenbankschemas, um die folgenden Ausgaben zu erhalten: [7 BE]
3. Eine Liste, absteigend nach Größe sortiert, die alle Gebäude mit Namen und Größe angibt, deren Gebäudenamen mit N oder O beginnen.
4. Eine Liste der Nummern aller Futtersorten und der jeweils zugehörigen Gesamtmenge, die von der jeweiligen Futtersorte von allen Tieren der Gattung "Gorilla“ an einem Tag konsumiert wird. Dabei sollen nur Futtersorten aufgeführt werden, bei denen diese Gesamtmenge größer als 100 (kg /pro Tag) ist.
5. Entwickeln Sie ein ER-Modell, das sich in das relationale Datenbankschema in Abbildung 1 überführen lässt. [6 BE]

Material

zu Aufgabenteil a)

|  |
| --- |
| **Person** |
| - benutzername: Zeichenkette  - passwort: Zeichenkette  - punkte: Ganzzahl  - gebDatum: Zeichenkette  - alter: Ganzzahl |
| c Person(n: Zeichenkette, p: Zeichenkette, a: Ganzzahl)  + getName(): Zeichenkette  + aenderePasswort(p: Zeichenkette)  + getAlter(): Ganzzahl  + getPunkte(): Ganzzahl  + setPunkte(n: Ganzzahl)  - aktualisiereAlter()  … |
| Abbildung 2: Klassenkarte der Klasse Person |

zu den Aufgabenteilen e) und f)



Abbildung 3: Struktogramm der rekursiven Operation sortiereNeu

Hinweis: Die Division von Ganzzahlen im Struktogramm ist als Division ohne Rest zu verstehen.

zu Aufgabenteil e)

|  |  |  |  |
| --- | --- | --- | --- |
| Selbstaufruf | Belegung der Reihung user | | |
| [0] | [1] | [2] |
| sortiereNeu(0,2) | p0: Person  (punkte=33) | p1: Person  (punkte=81) | p2: Person  (punkte=17) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Abbildung 4: Vorlage zur Dokumentation der rekursiven Selbstaufrufe  
mit Belegung der Reihung user zum Zeitpunkt des Aufrufs

zu Aufgabenteil g)



Abbildung 5: Beispielhafter Binärbaum heimischeVoegel

Hinweis: Grau eingefärbte Knoten sind Merkmale.

****

Abbildung 6: Struktogramm der Operation bestimmeVogel