

# 46116 Herbst 2017

Softwaretechnologie / Datenbanksysteme (nicht vertieft)

Aufgabenstellungen mit Lösungsvorschlägen



**Die Bschlangaul-Sammlung**

Hermine Bschlangaul and Friends

# Aufgabenübersicht

Thema Nr. 1	3
Teilaufgabe Nr. 1	3
Aufgabe 4 [Qualitätssicherung, Testen bei Bubblesort]	3
Thema Nr. 2	5
Teilaufgabe Nr. 1	5
Aufgabe 3: Entwurfsmuster [Entwurfsmuster bei Bankkonten, Hockey- spiel, Dateisystem]	5
Teilaufgabe Nr. 2	5
Aufgabe 4 [Turmspringen]	5



## Die Bschlangaul-Sammlung

Hermine Bsclangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

# Thema Nr. 1

## Teilaufgabe Nr. 1

### Aufgabe 4 [Qualitätssicherung, Testen bei Bubblesort]

Ein gängiger Ansatz zur Messung der Qualität von Software ist das automatisierte Testen von Programmen. Im Folgenden werden praktische Testmethoden anhand des nachstehend angegebenen Sortieralgorithmus diskutiert.

#### Algorithmus 1 Bubble Sort

```
public class BubbleSort {  
    void bubblesort(int[] array, int len) {  
        for (int i = 0; i < len - 1; i++) { // 1  
            for (int j = 0; j < len - 1; j++) { // 2  
                if (array[j] > array[j + 1]) { // 3  
                    int temp = array[j]; // 4  
                    array[j] = array[j + 1]; // 5  
                    array[j + 1] = temp; // 6  
                }  
            }  
        }  
    }  
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_46116/jahr\\_2017/herbst/BubbleSort.java](https://github.com/org/bschlangaul/examen/examen_46116/jahr_2017/herbst/BubbleSort.java)

- (a) Nennen Sie eine Art des Black-Box-Testens und beschreiben Sie deren Durchführung anhand des vorgegebenen Algorithmus.

Lösungsvorschlag

Beim Black-Box-Testen sind die Testfälle von Daten getrieben (Data-Driven) und beziehen sich auf die Anforderungen und das spezifizierte Verhalten.)

⇒ Aufruf der Methoden mit verschiedenen Eingangsparametern und Vergleich der erhaltenen Ergebnisse mit den erwarteten Ergebnissen.

Das Ziel ist dabei eine möglichst hohe Anforderungsüberdeckung, wobei man eine minimale Anzahl von Testfällen durch Äquivalenzklassenzerlegung (1) und Grenzwertanalyse (2) erhält.

**zu (1):** Man identifiziert Bereiche von Eingabewerten, die jeweils dieselben Ergebnisse liefern. Dies sind die sog. Äquivalenzklassen. Aus diesen wählt man nun je einen Repräsentanten und nutzen diesen für den Testfall.

**zu (2):** Bei der Grenzwertanalyse identifiziert man die Grenzbereiche der Eingabedaten und wählt Daten aus dem nahen Umfeld dieser für seine Testfälle.

Angewendet auf den gegebenen Bubblesort-Algorithmus würde die Grenzwertanalyse bedeuten, dass man ein bereits aufsteigend sortiertes Array und ein abstei-

gend sortiertes Array übergibt.

- (b) Zeichnen Sie ein mit Zeilennummern beschriftetes Kontrollflussdiagramm für den oben angegebenen Sortieralgorithmus.

Lösungsvorschlag

Zur Erinnerung: Eine im Code enthaltene Wiederholung mit for muss wie folgt im Kontrollflussgraphen „zerlegt“ werden:

- (c) Erklären Sie, ob eine vollständige Pfadüberdeckung für die gegebene Funktion möglich und sinnvoll ist.

Lösungsvorschlag

Eine vollständige Pfadüberdeckung ( $C_1$ -Test) kann nicht erreicht werden, da die Bedingung der inneren Wiederholung immer wahr ist, wenn die Bedingung der äußeren Wiederholung wahr ist. D. h., der Pfad S-1-1-2-2-1“ kann nie gegangen werden. Dies wäre aber auch nicht sinnvoll, weil jeder Eintrag mit jedem anderen verglichen werden soll und im Fall true  $\rightarrow$  false ein Durchgang ausgelassen.

# Thema Nr. 2

## Teilaufgabe Nr. 1

### Aufgabe 3: Entwurfsmuster [Entwurfsmuster bei Bankkonten, Hockeyspiel, Dateisystem]

Verwenden Sie geeignete **Entwurfsmuster**, um die folgenden Sachverhalte mit Hilfe von **UML-Klassendiagrammen** zu beschreiben. Nennen Sie das zu verwendende Entwurfsmuster namentlich, wenden Sie es zur Lösung der jeweiligen *Fragestellung* an und erstellen Sie damit das problemspezifische UML-Klassendiagramm. Beschränken Sie sich dabei auf die statische Sicht, definieren Sie keinerlei Verhalten mit Ausnahme der Definition geeigneter Operationen.

- (a) Es gibt unterschiedliche Arten von Bankkonten: Girokonto, Bausparkonto vmd Kreditkarte. Bei allen Konten ist der Name des Inhabers hinterlegt. Girokonten haben eine IBAN. Kreditkarten sind immer mit einem Girokonto verknüpft. Bei Bausparkonten werden ein Sparzins sowie ein Darlehenszins festgelegt. Es gibt eine *zentrale* Klasse, die die *Erzeugung* unterschiedlicher Typen von Bankkonten steuert.
- (b) Beim Ticker für ein Hockeyspiel können sich verschiedene Geräte registrieren und wieder abmelden, um auf *Veränderungen* des Spielstands zu *reagieren*. Hierzu werden im Ticker die Tore der Heim- vmd Gastmannschaft sowie die aktuelle Spielminute vermerkt. Als konkrete Geräte sind eine Smartphone-App sowie eine Stadionuhr bekannt.
- (c) Dateisysteme sind *baumartig* strukturiert. Verzeichnisse können wiederum selbst Verzeichnisse und/oder Dateien beinhalten. Sowohl Dateien als auch Verzeichnisse haben einen Namen. Das jeweilige Elternverzeichnis ist eindeutig. Bei Dateien wird die Art (Binär, Text oder andere) sowie die Größe in Byte, bei Verzeichnissen die Anzahl enthaltener Dateien hinterlegt.

## Teilaufgabe Nr. 2

### Aufgabe 4 [Turmspringen]

Für die bayerische Meisterschaft im Turmspringen ist folgendes Datenbankschema angelegt:

Springer : {[ Startnummer, Nachname, Vorname, Geburtsdatum, Körpergröße ]}

Sprung : {[ SID, Beschreibung, Schwierigkeit ]}

springt : {[ SID, Startnummer, Durchgang ]}

FK (SID) referenziert Sprung (SID)

FK (Startnummer) referenziert Springer (Startnummer)

Das Attribut Schwierigkeit kann die Werte 1 bis 10 annehmen, das Attribut Durchgang ist positiv und ganzzahlig. Die Körpergröße der Springer ist in Zentimeter angegeben.

- (a) Welche Springer sind größer als 1,80 m? Schreiben Sie eine SQL-Anweisung, welche in der Ausgabe mit dem größten Springer beginnt.

Lösungsvorschlag

```
SELECT Vorname, Nachname, Körpergröße
FROM Springer
WHERE Körpergröße > 180
ORDER BY Körpergröße DESC;
```

- (b) Welche Springer haben im ersten Durchgang einen Sprung mit einer Schwierigkeit von unter 6 gezeigt? Schreiben Sie eine SQL-Anweisung, welche Startnummer und Nachname dieser Springer ausgibt.

Lösungsvorschlag

```
SELECT Springer.Startnummer, Springer.Nachname
FROM Springer, Sprung, springt
WHERE
  Sprung.SID = springt.SID AND
  Springer.Startnummer = springt.Startnummer AND
  springt.Durchgang = 1 AND
  Sprung.Schwierigkeit < 6;
```

- (c) Formulieren Sie in Umgangssprache, aber trotzdem möglichst präzise, wonach mit folgender Abfrage gesucht wird:

```
SELECT springt.Startnummer, s.Nachname, s.Vorname, MAX(springt.Durchgang)
FROM springt, Springer s
WHERE springt.Startnummer = s.Startnummer
GROUP BY springt.Startnummer, s.Nachname, s.Vorname
```

Lösungsvorschlag

Die Abfrage gibt die Startnummer, den Nachnamen, den Vornamen und die Anzahl der Sprünge, d. h. die Anzahl der Durchgänge der einzelnen Springer an.

- (d) Gesucht ist die „durchschnittliche Körpergröße“ all der Springer, die vor dem 01.01.2000 geboren wurden. Formulieren Sie eine SQL-Anweisung, wobei die Spalte mit der durchschnittlichen Körpergröße genau diesen Namen „durchschnittliche Körpergröße“ haben soll.

Lösungsvorschlag

Umlaute und Leerzeichen sind bei Spaltenbeschriftungen nicht erlaubt.

```
SELECT AVG(Körpergröße) AS durchschnittliche_Koerpergroesse
FROM SPRINGER
WHERE Geburtsdatum < DATE('2000-01-01');
```

oder

```
SELECT AVG(Körpergröße) AS durchschnittliche_Koerpergroesse  
FROM SPRINGER  
WHERE Geburtsdatum < '01.01.2000';
```