Fragen, Antworten, Kommentare

Die Online-Befragung zur genutzten alternativen Veranstaltungsform und zur Lehrevaluation ist online. Bitte ausfüllen: https://forms.gle/nVYLgQbEQ1hpSU6HA. Sie werden eventuell aufgefordert sich bei Google anzumelden, das ist nur notwendig, wenn Sie in der Bearbeitung eine Pause machen wollen und das Teilergebnis zwischenspeichern wollen. Die Befragung endet am 20.12., die Ergebnisse stehen in einem nachfolgenden Fragen&Antworten-Dokument auf der Webseite der Veranstaltung.

Frage: Sollen wir zu Blatt 11 DBUnit- oder JUnit-Tests schreiben?

Antwort: Nein das ist nicht gefordert. Schreiben Sie passende SQL-Statements und kopieren Sie die zugehörigen Ergebnisse. Natürlich dürften Sie auch „ordentliche“ tests schreiben.

Frage: Da wir in der Vorlesung noch nicht auf NoSQL-Datenbankmanagementsystemen eingegangen sind, wollte ich nochmal nachfragen wie es dort mit der Bewertung aussieht. Da wir uns selbständig in dieses Thema einarbeiten müssten wollte ich nachfragen ob dies auch in der Bewertung berücksichtigt wird oder wird die selbständige Einarbeitungszeit in das Thema nicht berücksichtigt und beides wird komplett gleich gewertet.

Klar ist, dass eine NoSQL-Datenbank eine höhere Einarbeitungszeit hat, da erst das generelle Konzept verstanden werden muss. Dies wird auch im Bericht deutlich, da das Kapitel zu den „Grundlagen“ deutlich länger wird, da das Konzept vorgestellt werden muss. Die nachfolgenden Kapitel sind dann teilweise kürzer oder nur ein Satz, der begründet, warum das Thema nicht zum gewählten Datenbanktyp passt, die Analyse der Anfragemöglichkeiten ist ähnlich aufwändig. Zu beachten ist aber, dass auch relationale Datenbanken einiges an Einarbeitung kosten, sei es um zusätzliche Werkzeuge zu studieren, genau auf die Datentypen zu schauen, die Mächtigkeit der unterstützen SQL-Anfragen zu analysieren, die Transaktionssteuerung zu betrachten und ob und wie Trigger umgesetzt werden zu studieren. Für sehr gute Ergebnisse schätze ich den Arbeitsaufwand ähnlich ein.

Hinweis: In einer der letzten Praktikumsaufgaben wurde der Transaktionsbegriff an praktischen Beispielen experimentell kennengelernt. Das Thema wir am Ende der Vorlesung aufgegriffen. Es sei angemerkt, dass es sich bei *Transaktionen* um einen der wichtigsten, wohl den wichtigsten Begriff der Datenbankveranstaltung handelt. Das Thema, dass mehrere Prozesse sich fachlich überlappen, wird auch in anderen Veranstaltungen behandelt. „Spoiler Alert“: Es gibt nicht die immer passende Lösung, sondern man muss wissen warum eine Lösung die passendste zur jeweiligen Aufgabenstellung ist.

Frage: Soll in Java eine gefangene Exception weitergegeben werden?

Antwort: Die sauberste Form ist es die Exception zu fangen, ihren Grund zu verstehen und eine projekteigene Exception mit dem gefundenen Grund dann weiterzureichen, wenn die Exception nicht lokal bearbeitet werden kann. Eine direkte Weitergabe ist denkbar, wenn nur eine der aufrufenden

Methoden die Exception fachlich behandeln kann. Eine durchaus interessante Variante ist es, statt einer Exception ein neues Ergebnisobjekt einzuführen. Dies hat z. B. die Objektvariablen ergebnis vom erwarteten Ergebnis-Typ, ein boolesches Flag, ob das Ergebnis ok ist und einen Text mit einer Fehlerinformation, falls das Ergebnis nicht ok ist. Natürlich könnten weitere Informationen im Ergebnis dann vorhanden sein. Der Vorteil ist, dass immer ein sinnvolles Ergebnis zurückgeliefert wird und bei der Nutzung dieser Methode immer das Ergebnis auf Korrektheit überprüft werden muss, da seine Struktur bekannt ist

Frage: Meine Methode liefert ein int als Ergebnis, bei einer SQLException gebe ich einen Dummy-Wert zurück, ist das ok?

Antwort: Nein, Es besteht immer die Gefahr, dass der Dummy-Wert als normales Ergebnis behandelt wird und so in die Verarbeitung kommt. Der Trick Integer zurückzugeben und dann als Ergebnis null zu liefern ist auch schmuddelig. Sauber ist das Ergebnis auf die vorherige Frage, ein eigenes Ergebnis-Objekt zu liefern. Ein gutes Beispiel ist die generische Java-Klasse Optional<> die z. B. als Optional<Integer> einen ganz normalen Integer-Wert als value haben kann, die aber auch eine Methode zum Prüfen hat, ob das Ergebnis ok ist. Statt eines null-Wertes wird dann Optional.empty() zurückgegeben (https://www.baeldung.com/java-optional). Erhält eine aufrufende Methode ein Optional-Objekt ist diese dafür verantwortlich das Objekt zu überprüfen.