Freie Universität Berlin, Institut für Informatik, Arbeitsgruppe Software Engineering
Prof. Dr. L. Prechelt, Franz Zieris, Michaela Borzechowski, Tanja Linkermann, Oliver Wiese

Übungsblatt 11 **Analytische Qualitätssicherung 2** zum Tutorium in KW 27

Bereiten Sie Ihre **Lösungen** grundsätzlich so vor, dass Sie diese in der Übung Ihren Kommilitonen in geeigneter Form **zeigen** und **diskutieren** können. Geben Sie bitte stets Ihre verwendeten **Quellen** an.

Aufgabe 11-1: Durchsichten

Auf der nächsten Seite finden Sie eine Checkliste zur Durchsicht von Java-Code.

1. Gegen welche der dort aufgeführten Prüfpunkte (*checks*) verstoßen Sie selbst gelegentlich bei Ihrer Programmierarbeit?
2. Nennen Sie mindestens einen Punkt, der nicht automatisiert geprüft werden kann und erklären Sie, warum das nicht geht.
3. Nennen Sie mindestens einen Punkt, der Defekte aufdeckt, deren potenziellen Auswirkungen (= Versagensfälle) durch Testen schwer zu entdecken sind und erläutern Sie diesen.
4. Welche Vorteile haben Durchsichten im Allgemeinen im Vergleich zu dynamischen Tests? Beschreiben Sie mindestens drei solcher Vorteile.

Aufgabe 11-2★: Testtechniken anwenden und bewerten

1968 hat Edsger Dijkstra eine kurze, aber berühmt gewordene Notiz veröffentlicht über die Gefährlichkeit von Sprunganweisungen in Programmiersprachen: „Goto considered harmful". Eine schnelle Suche wird Sie diesen Artikel schnell finden lassen. Lesen Sie ihn und geben Sie die Argumentation in groben Zügen wieder. Stimmen Sie ihr zu und finden Sie sie überzeugend?

Checkliste zur Durchsicht von Java-Programmen

Arithmetik

1. Sind Überlauf oder Unterlauf während der Berechnung möglich?
2. Für jeden Ausdruck mit mehr als einem Operator: Sind die Annahmen über die Ausführungsreihenfolge korrekt?
3. Wurden Klammern eingesetzt um Mehrdeutigkeit zu vermeiden?
4. Ist „Division by Zero" möglich?
5. Geht man fälschlicherweise davon aus, dass Fließkomma-Arithmetik genau ist?

Schleifen

6. Werden vor einer Schleife alle beteiligten Variablen richtig initialisiert oder enthalten sie alle bereits einen gültigen Wert?
7. Werden alle Schleifen auf jeden Fall beendet?

Abzweigungen

8. Hat jedes `switch`-Statement einen `default`-Fall?
9. Sind fehlende `break`-Statements in `switch`-Blöcken korrekt und gesondert kommentiert?
10. Sind alle `else`-Zweige richtig behandelt? Wenn einer `if`-Bedingung der `else`-Zweig fehlt, wird der Fall richtig behandelt, wenn die `if`-Bedingung nicht erfüllt wird?

Datenfluss

11. Kann eine Variable unter Umständen `null` sein und wird dieser Fall abgefangen?
12. Werden Parameterwerte auf gültige Wertebereiche (entsprechend der Vorbedingungen) überprüft?
13. Müssen Objekte mit `equals()` oder direkt mit `==` verglichen werden?
14. Kann eine Typanpassung (casting) fehlschlagen?

Arrays

15. Ist das Indexieren von Arrays außerhalb des gültigen Bereichs möglich?

Funktionsaufrufe

16. Reagiert der Aufrufer einer Methode auf alle möglichen Werte, die zurückgegeben werden können, inkl. Ausnahmen (Exceptions)?
17. Erfüllt der Aufrufer einer Methode die Voraussetzungen für die Parameter der Methode?
18. Kann ein Stack-Overflow bei rekursiven Funktionen auftreten?

Multithreading

19. Sind alle Zugriffe von mehreren Threads auf dieselben Variablen synchronisiert? Muss die Variable als `volatile` deklariert werden?
20. Besteht eine Verklemmungsgefahr?
21. Werden Threads, die mit `wait()` warten, irgendwann mal aufgeweckt?
22. Werden `InterruptedExceptions` behandelt?
23. Erfolgt der Zugriff auf eine Variable in zwei getrennten `synchronized`-Blöcke vom selben Thread und ist das korrekt?
24. Wird ein Objekt von einem Thread freigegeben (auf `null` gesetzt) und von einem anderen zugegriffen?

Ausnahmebehandlung

25. Wird der Kontrollfluss richtig fortgesetzt, wenn eine Ausnahme auftritt oder abgefangen wird?
26. Werden eventuelle Ausnahmen beim Zugriff auf externe Ressourcen abgefangen?
27. Werden alle Ausnahmen, die in den von dieser Methode aufgerufenen Methoden aufgeworfen werden können, abgefangen?

Externe Ressourcen

28. Werden Eingabe-, Ausgabe-Ströme und externe Ressourcen (Datenbank-Connections, Sockets etc.) wieder geschlossen?
29. Werden gepufferte Daten "geflusht"?
30. Werden Ausnahmen (Exceptions) bei Ein- und Ausgabe richtig behandelt?