PHPmagazin

Veröffentlicht auf *JAXenter.de* (http://jaxenter.de)

Automatisiertes Testen von JavaFX GUI-Komponenten

Autor:

Wolfgang Weigend

Seit März 2014 ist JavaFX ein fester Bestandteil des JDK 8. JavaFX 8 soll künftig auch zur Realisierung von kritischen Geschäftsanwendungen verwendet werden können. Dazu ist es notwendig, dass einerseits der Reifegrad der neuen JavaFX-UI-Technologie akzeptabel ist und andererseits die Benutzeroberflächen von Geschäftsanwendungen auf ihre korrekte Anwendungsfunktionalität ausgiebig getestet werden können. Der Schwerpunkt liegt auf einem hohen Datenvolumen, das der grafischen Benutzeroberfläche mit automatisierten Tests zugeführt wird. Wir betrachten im Folgenden, wie ein solches Testszenario mit dem Werkzeug QF-Test aufgebaut werden kann.

[1]

Testen von JavaFX-Anwendungen mit Werkzeugunterstützung

Bei der Entwicklung von Anwendungen spielt das Testen eine große Rolle. Nach Aussage des aktuellen World Quality Reports entfällt darauf bereits ein Viertel des IT Budgets. Mit steigender Komplexität bei der Anwendungsentwicklung erhöhen sich auch die Test-Anforderungen. Eine möglichst umfassende Testabdeckung kann durch

Automatisierung erreicht werden.

Mit geeigneten Werkzeugen kann das Erstellen der automatisierten Tests vereinfacht und eine hohe Integration in die Entwicklung gewährleistet werden. Automatisierte UI-Tests haben dabei den Vorteil, dass sie im Gegensatz zu Unit-Tests nicht nur eine isolierte Einheit, sondern einen ganzen Prozess mit einem Test abdecken können. Durch Einsatz der richtigen Testumgebung ist es möglich, Tests zum Großteil von Fachtestern erstellen zu lassen und somit mehr Ressourcen für die Entwicklung zur Verfügung zu haben. Dabei sollte es die Testumgebung dem Tester ermöglichen, mit vertrauten Begriffen und Objekten zu arbeiten.

Gerade bei Oberflächentests ist es wichtig, dass der Tester mit den Objekten arbeitet, die er kennt, und sie auf der Oberfläche der Anwendung wiederfindet, auch wenn der tatsächliche Aufbau der GUI viel komplexer ist. Bereits bei einfachen Java-FX-Anwendungen besteht die Oberfläche aus vielen einzelnen Elementen in einer tief verschachtelten Baumstruktur. **Abbildung 1** zeigt eine Beispielanwendung zur Konfiguration von Neufahrzeugen.

Abbildung 1

Die Struktur der Anwendungsoberfläche wird in einer 3D-Ansicht dargestellt (**Abbildung 2**), welche die Komplexität ihrer Verschachtelung plastisch hervorhebt. Für das Testen ist jedoch nur ein kleiner Teil der Elemente interessant, alle anderen Elemente sind auf technischer Ebene eingebunden. Das Testwerkzeug ermöglicht die Reduktion dieser komplexen Oberflächen auf das Wesentliche.

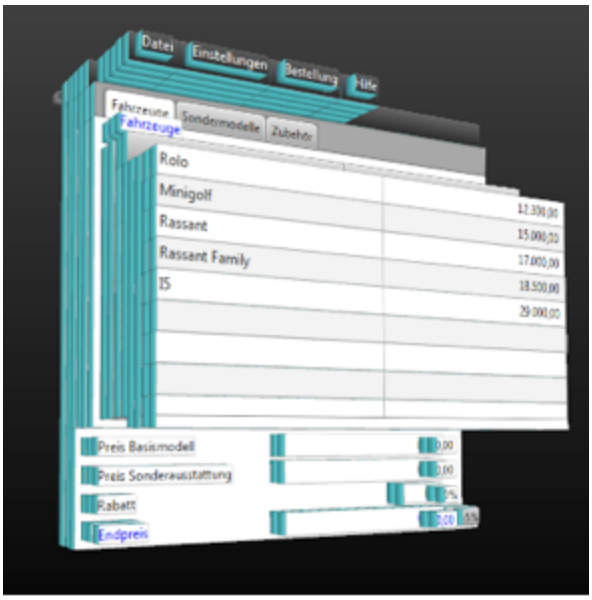

Abbildung 2

Abbildung 3 zeigt die vereinfachte Struktur, so wie sie das Testwerkzeug dem Anwendungstester zur Verfügung tellt.

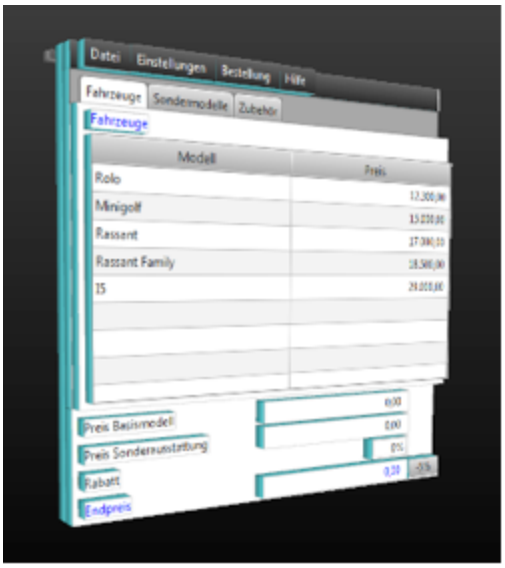

Abbildung 3

Für Sonderfälle kann mit der vollständigen Hierarchie gearbeitet werden, und entwicklernahe Tests können durch Skripte durchgeführt werden. Mittels Vereinfachung und Generalisierung der UI-Komponenten führt die Unterstützung eines modularen Testaufbaus dazu, dass Testbestandteile wiederverwendet werden können. Dadurch sinkt der Aufwand bei der Erstellung neuer Tests, und die Wartbarkeit erhöht sich. Auf dieser Basis können Module oder Bibliotheken für die Fachtester erstellt werden. Durch die zusätzliche Integration von Datentreibern wird datengetriebenes Testen auch auf der Oberfläche ermöglicht, und Massentests können realisiert werden.

Fazit

Anwendungsoberflächentests sind für die Anwendungsentwicklung unerlässlich. Durch die direkte Integration im Entwicklungsprozess kann damit ein Continuous-Integration-Szenario aufgebaut werden. Testwerkzeuge erleichtern das Testen von JavaFX-8-Anwendungen durch Vereinfachung der Komponentenhierarchie und ihren modularen Aufbau und sind damit auch für Fachtester ohne Programmierkenntnisse nutzbar.

Aufmacherbild: Vector customer feedback concept in flat style [2] von Shutterstock / Urheberrecht: venimo

Links & Literatur:
http://docs.oracle.com/javase/8/javase-clienttechnologies.htm [3]
http://docs.oracle.com/javase/8/javafx/api/toc.htm [4]
http://www.capgemini.com/thought-leadership/world-quality-report-2014-15 [5]
http://www.qfs.de/de/qftest/index.html [6]

Tagline:
JavaFX 8 UI-Anwendungsfunktionalität testen

Links:
[1] http://jaxenter.de/sites/default/files/preview_images/javafx_anwendungen_testen.png
[2] http://www.shutterstock.com/pic-179555099/stock-vector-vector-customer-feedback-concept-in-flat-style-hand-checking-excellent-mark-in-a-survey.html?src=kV4wQz1KX7wUDyWmVt7ZEw-1-2
[3] http://docs.oracle.com/javase/8/javase-clienttechnologies.htm
[4] http://docs.oracle.com/javase/8/javafx/api/toc.htm
[5] http://www.capgemini.com/thought-leadership/world-quality-report-2014-15
[6] http://www.qfs.de/de/qftest/index.html