

## Java-Entwicklung

# JavaFX – Das neue Gesicht für Java-Anwendungen

Bislang konnte Java nur eingeschränkt mit ansprechenden Oberflächen glänzen. Das ändert sich mit JavaFX, das für eine große Anzahl an Plattformen verfügbar sein wird. Java-Anwendungen werden künftig auf Desktops, Smartphones und Embedded Geräten eine gute Figur machen und eine gute User Experience bieten.

Obwohl Java von Anfang an mit Oberflächenfunktionen wie AWT und Swing ausgestattet war, konnte sich diese Technologie nicht auf dem Desktop durchsetzen. Wenig hilfreich ist in diesem Zusammenhang, dass Oracle die Oberflächenbibliotheken nicht mehr weiterentwickelte und nur über externe Software zeitgemäße Oberflächen erstellt werden können. Etwas anders sieht es im Bereich Web-Oberflächen aus. Hier wird Java zumindest auf dem Server sehr oft eingesetzt und es existiert eine große Anzahl von Web-Frameworks.

Die weitere Verbreitung von mobilen und Embedded Geräten stellt ganz neue Anforderungen an die Gestaltung und Bedienung von Applikationen. Da Java gerade für diese Bereiche ursprünglich entworfen wurde, ist zu begrüßen, dass Oracle mit JavaFX eine neue Plattform für moderne Rich Client-Oberflächen anbietet. JavaFX wird mit der kommenden JDK 8 Version ausgeliefert. Für Entwickler vereinfacht sich dadurch der Entwurf von Rich Client-Anwendungen für unterschiedliche Plattformen. Mit dem JavaFX Packager lässt sich die Ablaufumgebung gleich integrieren, so dass der Anwender gar nicht merkt, dass er mit einer Java-Anwendung arbeitet. Selbst so kostengünstige Umgebungen wie der Einplatinen-Computer Raspberry Pi werden hiermit unterstützt.

## Warten auf die große Acht

Die Versuche von Sun mit JavaFX Script waren wenig erfolgversprechend. Doch seit JavaFX Bestandteil der Java 6 Ablaufumgebung ist und auch nachfolgende Versionen deutlichen Mehrwert bieten, besteht Hoffnung, dass sich Java wieder stärker bei Client-Oberflächen durchsetzt – vor allem im Embedded-Bereich. Als Endgeräte kommen hier



beispielsweise Smartphones, Kiosk-Systeme, Set-Top-Boxen, Einplatinen-Computer, Desktop-Computer oder Blu-ray-Player in Frage.

Da eine Migration bestehender Oberflächen eher schwierig ist, bietet sich der Einsatz für Neuprojekte an. Es gibt jedoch mit SwingNode einen Migrationspfad, Swing-Anwendungen schrittweise auf JavaFX umzustellen. Da das mittlerweile veraltete Swing-Toolkit ebenfalls nicht mehr weiterentwickelt wird, ist die Migration insbesondere für weiterhin langläufige Projekte ein interessanter und gangbarer Ansatz. Vor allem, wenn man auf dem Markt weiterhin mit modernen Tools konkurrieren möchte, sollte man auf einige Funktionen von JavaFX nicht verzichten.

Mobile Anwendungen können über WebView-Node als Anwendungen mit Web-Technologien entwickelt und in JavaFX-Anwendungen integriert werden. Dabei ist es von Vorteil, dass JavaFX CSS (Cascading Style Sheets) zur grafischen Gestaltung verwendet und für die Web-Darstellung die WebKit Rendering Engine einsetzt. Diese wird unter anderem auch von Google Chrome genutzt. Bei Bedarf lassen sich bei diesem hybriden Ansatz dennoch die leistungsfähigeren nativen Funktionen nutzen. Durch dieses Konzept ist es beispielsweise möglich, interaktive Google Maps Darstellungen in Desktop-Anwendungen zu integrieren.

Neben WebView gibt es noch weitere Komponenten, die über die allgemein üblichen Darstellungsformen wie Tabellen und Buttons hinausgehen:

Mit MediaView lassen sich verschiedenste Formate wie Videos oder Sound-Dateien nahtlos in eine JavaFX-Anwendung integrieren. Zusätzlich bietet JavaFX mehrere Komponenten zur Darstellung von Business-Diagrammen an. Diese können leicht durch die angebotenen Funktionen angepasst und sogar über CSS gestaltet werden.

Mit Java 8 bekommt JavaFX auch eine volle 3D-Unterstützung. Die 3D-Daten-Visualisierung wird so deutlich erleichtert, um beispielsweise Lagepläne oder komplexe Diagramme übersichtlicher darzustellen. Hierbei unterstützt JavaFX nicht nur 3D-Modelle, sondern auch dynamische Lichtquellen und verschiedene Texturformate, wie sie auch in modernen 3D-Anwendungen zum Einsatz kommen.

Mit der Version 8 wird JavaFX komplett im Java-Standard integriert und alle relevanten Bestandteile für Business- und Multimedia-Anwendungen sind vorhanden und erprobt. Interessierte Entwickler können sich das Ganze aber bereits heute über die Preview-Version von Java 8 anschauen. Die finale Version soll Anfang 2014 erscheinen.

## JavaFX goes Enterprise

Der JavaFX-Nutzung in Geschäftsapplikationen steht künftig nichts mehr im Wege. Eine sehr gute Werkzeugunterstützung

sowie moderne Funktionen erlauben es den Entwicklern, deutlich einfacher intuitive und ansehnliche Oberflächen zu erstellen. Spezielle UI- und Workflow-Wünsche des Kunden lassen sich damit einfacher und eleganter umsetzen. Folgende Beispiele zeigen, welchen Nutzen Entwickler und Anwender von JavaFX haben.

Das GUI-Komponenten-Toolkit von JavaFX ist so modular aufgebaut, dass es jederzeit um neue Komponenten oder aktualisierte Erscheinungsformen von vorhandenen Komponenten erweitert werden kann. Beispiele hierfür lassen sich unter anderem in der Enzo-Bibliothek finden, die JavaFX um mehrere grafische Module zur Visualisierung von Daten erweitert. Als Vorlage dienen hier verschiedenste Messgeräte.

Ein weiteres Beispiel ist AquaFX, das ein auf Apple Mac OS basiertes Design für alle Standardkomponenten von JavaFX bereitstellt. Durch Nutzung dieser Bibliothek ist eine JavaFX-Anwendung nicht mehr von einer nativen Mac-Anwendung unterscheidbar. Eine noch bessere Integration in das Betriebssystem des Kunden ist durch Nutzung nativer Installer erreichbar. Durch ein Build Tool von Oracle ist jedes JavaFX-Programm als native Anwendung auslieferbar. Unter Windows bekommt der Kunde so zum Beispiel einen eigenen Installer geliefert, der mithilfe von Wizard-Dialogen durch die Installation der Anwendung leitet. Hierbei kann man keinen Unterschied mehr zwischen der Java-Applikation und einer reinen Windows-Anwendung (Office, etc.) feststellen. Ein großer Vorteil ist, dass die unterlagerte Java-Umgebung fester Bestandteil der nativen Anwendung ist und eine vorherige Java-Installation bzw. nachgelagerte Administration so komplett entfällt. Die ersten JavaFX-Programme sind sogar bereits im Mac AppStore von Apple zu finden. Dieser Spezialfall funktioniert momentan zwar noch über Umwege, wird aber mit hoher Wahrscheinlichkeit zum Erscheinen von Java 8 noch deutlich verbessert und ausgebaut.

Neben den klassischen Desktop-Rechnern ist JavaFX auch auf zahlreichen anderen Plattformen lauffähig. Durch die Unterstützung von Embedded Devices und dem ARM-Chipsatz ist JavaFX auf kleinster Hardware ausführbar und bietet so beispielsweise eine sehr elegante Lösung für Kiosk-Systeme und Home Automation. Vor allem in diesen Bereichen war man bisher gezwungen, auf elegante oder hochauflösende grafische Darstellungen zu verzichten oder direkt auf ein PC-System, z.B. in der Form eines „kleinen“ Cubes, zurückzugreifen. Durch die in den vergangenen Jahren deutlich gestiegene Performance von ARM-Prozessoren und die neue Möglichkeit, JavaFX als Oberflächensprache auf diesen ablaufen zu lassen, ergeben sich jetzt völlig neue Möglichkeiten. Allein die Kosten- und Stromersparnis im Bereich der Hardware ist enorm.

Auch für moderne Arbeitsgeräte wie dem Windows Surface mit WindowsRT ist eine komplette Integration vorhanden. Gerade auf diesen Geräten zeigen sich weitere Stärken wie die



COMING  
SOON

MultiTouch-Fähigkeit von JavaFX. Zoom- und Swipe-Gesten, wie sie beim iPad vorkommen, werden von JavaFX voll unterstützt und sind leicht umzusetzen. Hierdurch ist endlich ein Einzug dieser neuen Interaktionsform in moderne Business-Anwendungen möglich. Wer hat sich schließlich nicht schon oft gewünscht, durch große Diagramme, Grafiken oder Lagepläne bequem mit Hilfe von einfachen Touch-Gesten zu navigieren?

Aber auch das bereits angesprochene iPad bleibt nicht unbeachtet. Zwar sind sowohl iOS als auch Android bisher noch keine offiziell unterstützten Plattformen für JavaFX, aber das wird sicherlich nicht mehr lange dauern. So ist es beispielsweise erst kürzlich gelungen, JavaFX-Anwendungen mit Hilfe der Open Source-Lösung RoboVM auf einem iPad zu betreiben.

Neben der hier beschriebenen UI-Gestaltung und den Interaktionen bietet das JavaFX-Ökosystem noch eine Fülle weiterer wichtiger Funktionen und Werkzeuge. Durch die konsequente Nutzung von CSS und die hierdurch mögliche Trennung von Entwickler- und Design-Teams wird der Entwicklungsprozess deutlich vereinfacht. Als Schnittmenge zwischen diesen beiden Instanzen kann der Scene Builder genutzt werden, mit dem sich Dialog-Layouts so einfach wie in einem modernen Publisher-Programm erstellen lassen. Während die Designer diese Layouts ausrichten und gestalten, kann das gesamte Entwicklerteam bereits eine An- und Einbindung dieser Dialoge in das Programm vornehmen. Für eine saubere und schnelle Anbindung an vorhandene Enterprise- und SOA-Systeme ist auch gesorgt. Projekte wie DataFX und OpenDolphin nehmen Entwicklern hierbei einen Großteil der Arbeit bereits ab und erlauben eine Integration und Darstellung von Web-Services-Daten.

## Fazit

JavaFX bringt Java bei modernen Oberflächen wieder ins Spiel und verbindet Desktop, Mobile, Embedded und Web auf eine einzigartige Weise. Spätestens mit dem Erscheinen von Java 8 im nächsten Jahr wird JavaFX die gewünschte Aufmerksamkeit und Verbreitung erlangen. Durch die wöchentlich erscheinenden Previews von Java 8 kann mit der Umsetzung von größeren Projekten bereits heute begonnen werden. Aufgrund der breiten Plattformbasis und der modernen UI-Funktionen von JavaFX lohnt sich ein zweiter Blick für Unternehmen, die alte Oberflächen ablösen oder auf mobile Umgebungen umstellen wollen.

Als Java-Nutzer der ersten Stunde arbeitet Materna aktiv an der Weiter- und Neuentwicklung von Komponenten und Architekturmustern für JavaFX mit. ■

## Die Autoren



Frank Pientka ist Senior Softwarearchitekt bei Materna und Gründungsmitglied im iSAQB-Board. Er ist seit mehreren Jahrzehnten im Bereich Java EE tätig. Seine Schwerpunkte sind E-Commerce- und Portalanwendungen. Der Dipl.-Informatiker unterstützt Kunden mit Architektur-Reviews und Coachings, die richtigen Entscheidungen zu treffen und nachhaltig umzusetzen. Darüber hinaus gibt er sein Wissen in Form von Publikationen und Vorträgen weiter.



Hendrik Ebberts ist als Senior Java-Architekt bei Materna für die Bereiche Java-Architektur und DevOps zuständig. Er hat mehrere Jahre Erfahrung in der Entwicklung von Java-Anwendungen und -Bibliotheken. Sein Hauptinteresse liegt hierbei in den Bereichen UI-Toolkits, Middleware und DevOps. Hendrik Ebberts leitet als aktives Mitglied der Java-Community die Java User Group Dortmund und spricht auf internationalen Konferenzen und User Groups. Zusätzlich veröffentlicht er Fachartikel in Print- und Online-Medien. Im Bereich JavaFX leitet Hendrik Ebberts die Entwicklung der beiden Open Source-Projekte AquaFX und MarvinFX und beteiligt sich aktiv an einigen der „Top-Level-Projekte“ in diesem Bereich, wie beispielsweise DataFX, ControlsFX und JFXtras.