



Anton Epple hat mehr als fünfzehn Jahre Erfahrung in der Leitung von Java-Projekten und veröffentlichte zahlreiche Artikel über das Thema. Er ist weltweit als Berater für eine Vielzahl von Unternehmen tätig, angefangen bei Start-ups bis hin zu Fortune-500-Unternehmen. In seiner Freizeit organisiert er Events für die Java-Community. Anton ist ein Mitglied des NetBeans Dream Team, 2013 wurde er zum Java Champion und JavaOne Rockstar ernannt, und 2014 wurde ihm für seine Arbeit am DukeScript-Framework der Duke's Choice Award verliehen.

Anton Epple

JavaFX 8

Grundlagen und fortgeschrittene Techniken



dpunkt.verlag

Anton Epple
toni.epple@eppleton.de

Lektorat: René Schönfeldt
Copy-Editing: Ursula Zimpfer, Herrenberg
Herstellung: Frank Heidt
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:
Buch 978-3-86490-169-0
PDF 978-3-86491-649-6
ePub 978-3-86491-650-2

1. Auflage 2015
Copyright © 2015 dpunkt.verlag GmbH
Wiebinger Weg 17
69123 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Inhaltsverzeichnis

Vorwort	1
1 Workshop: Eine erste Anwendung erstellen	3
1.1 Ein JavaFX-Projekt mit der Entwicklungsumgebung anlegen	3
1.1.1 Wie erstelle ich ein JavaFX-Projekt in NetBeans?	3
1.1.2 Wie erstelle ich ein JavaFX-Projekt mit IDEA?	4
1.1.3 Wie erstelle ich ein JavaFX-Projekt mit Eclipse?	4
1.1.4 Ein JavaFX-Projekt mit Maven erstellen	5
1.1.5 Hallo Welt!	6
2 Wie ist eine JavaFX-Anwendung aufgebaut?	9
2.1 Anfang und Ende – die Application	9
2.1.1 Lebenszyklus einer JavaFX-Anwendung	9
2.1.2 Aufrufparameter auslesen	10
2.2 Vorhang auf – die Stage	11
2.2.1 Dialoge anzeigen	12
2.2.2 Darstellung der Stage verändern	12
2.2.3 Die Scene – Was wird gespielt?	13
2.2.4 Nodes – die Darsteller	13
2.3 Wie werden Layouts verwendet?	14
2.4 Workshop: Einen Twitter-Client erstellen	14
2.4.1 Vorbereitung	15
2.4.2 Die Anwendung erzeugen	15
2.4.3 Mit Twitter4j auf die Timeline zugreifen	16
3 Properties und Bindings	19
3.1 Beans und Properties	19
3.1.1 Klassische JavaBean-Properties	19
3.1.2 Die neuen JavaFX-Properties	20
3.1.3 Was sind die wichtigsten Klassen und Interfaces?	21

3.1.4	Wie legt man Properties an?	23
3.1.5	Wie findet man die Bean zu einer Property?	23
3.1.6	Wie werden Properties schreibgeschützt?	24
3.2	Wie verwendet man Bindings?	25
3.2.1	High-Level- und Low-Level-APIs	26
3.2.2	Wie können Bindings logisch verknüpft werden?	28
3.2.3	Was für Berechnungen sind mit numerischen Bindings möglich?	29
3.2.4	Wie verwendet man StringProperties in Bindings?	29
3.2.5	Was passiert, wenn wir den Wert einer gebundenen Property setzen?	30
3.2.6	Bidirektionale Bindings	30
3.2.7	Wenn, dann und ansonsten – noch mehr Bindings-Logik ..	31
3.3	Mit Listnern auf Änderungen reagieren	32
3.3.1	InvalidationListener	32
3.3.2	ChangeListener	33
3.3.3	Welchen Listener soll ich verwenden?	33
3.4	JavaFX Collections	36
3.4.1	Welche sind die wichtigsten Klassen und Interfaces?	36
3.4.2	Wie erzeuge ich eine JavaFX Collection?	36
3.4.3	Wie reagiere ich auf Änderungen?	36
3.4.4	Wie verwende ich Bindings mit ObservableList?	38
3.4.5	Wie observiere ich einen bestimmten Wert einer Collection? ..	38
3.4.6	Wie erzeuge ich eine Lazy ObservableList?	39
3.5	Properties und Performance	41
3.6	Grenzen der Verwendung von Properties	42
4	Welche Bedienelemente gibt es?	43
4.1	Welche Controls stehen zur Verfügung?	43
4.2	Wie werden Controls verwendet?	45
4.2.1	Wie werden Labels verwendet?	45
4.2.2	Tooltip	48
4.2.3	Buttons	49
4.2.4	Texteingabe mit TextField, PasswordField, TextArea und HTML-Editor	54
4.2.5	Numerische Werte mit dem Slider eingeben	58
4.2.6	Menüs und Kontextmenüs erstellen	60
4.2.7	ListView	65
4.2.8	TableView	75
4.2.9	TreeView und TreeTableView	78
4.2.10	Auswahlfelder	82

4.3	Controls fürs Layout verwenden	87
4.3.1	Accordion-Control und TitledPane verwenden	87
4.3.2	Mehrere Seiten mit der TabPane verwalten	88
4.3.3	Blättern mit Touchgesten und dem Pagination-Control	89
4.3.4	SplitPane	90
4.4	Workshop: Eine ListView für die Twitter-App	94
5	Ein Layout erstellen	97
5.1	Die eingebauten Layouts verwenden	97
5.1.1	VBox und HBox	97
5.2	Die BorderPane verwenden	99
5.3	Layouts mit der AnchorPane erstellen	100
5.4	Die FlowPane verwenden	100
5.5	Layout mit der StackPane	101
5.6	Layout mit der TilePane	103
5.7	Layout mit der GridPane	104
5.7.1	Wie füge ich Nodes hinzu?	104
5.7.2	Wie passt man Höhe und Breite der Columns und Rows an?	106
5.7.3	Wie werden einzelne Elemente ausgerichtet?	106
5.8	Unmanaged Nodes	107
5.9	Eigene Layoutcontainer erstellen	108
5.10	Workshop: Ein bestimmtes Layout umsetzen	110
6	Formulare mit FXML erstellen	117
6.1	Wozu FXML?	117
6.2	Beispielprojekt	118
6.3	Layouts per Drag-and-Drop mit dem SceneBuilder erstellen	119
6.3.1	Wie wird der SceneBuilder installiert?	119
6.3.2	Wie integriere ich den SceneBuilder mit NetBeans?	120
6.3.3	Wie verwende ich den SceneBuilder mit e(fx)clipse?	120
6.3.4	Wie konfiguriere ich IntelliJ IDEA?	121
6.3.5	Wie erstelle ich ein Formular?	121
6.4	Das FXML-Format verstehen	123
6.5	Daten an den Controller anbinden	124
6.5.1	Member-Variablen annotieren	125
6.5.2	Sonderfälle ResourceBundle und URL	125
6.5.3	Methoden annotieren	126
6.5.4	Sonderfall initialize-Methode	126
6.6	FXML-Dateien laden	126
6.7	FXMLLoader und Internationalisierung	126

6.8	Eigene Nodes verwenden	126
6.9	Workshop: Einen Taschenrechner erstellen	128
6.10	Fazit	134
7	Den Look mit CSS gestalten	135
7.1	Beispielanwendung	135
7.2	CSS Basics	137
7.2.1	Selektoren	137
7.2.2	Eigenschaften und Deklarationen	143
7.2.3	Eigenschaftsvererbung	144
7.2.4	Zusammenfassende Eigenschaften	145
7.2.5	Namenskonventionen	145
7.2.6	Die wichtigsten stylebaren Eigenschaften	146
7.2.7	Stylebare Eigenschaften eines Nodes	146
7.2.8	Shapes	146
7.2.9	Region und Controls	147
7.2.10	Verwendung im Beispiel	150
7.2.11	Besonderheiten	151
7.2.12	UserAgent-Stylesheet	151
7.2.13	Stylesheet einer Scene	152
7.2.14	Stylesheet eines Parent	152
7.2.15	Eigenschaften über die Java-API setzen	152
7.2.16	Inline-Styles	152
7.2.17	CSS-API	153
7.3	Workshop: Eine Anwendung stylen	154
7.3.1	Wie registriere ich das Stylesheet?	154
7.3.2	Wie definiere ich Variablen?	155
7.3.3	Wie style ich den Hintergrund?	155
7.3.4	Wie style ich den Home-Button?	156
7.3.5	Wie reagiere ich im Style auf Buttonklicks?	156
7.3.6	Wie style ich den StatusView?	157
7.4	Workshop: Eine Anwendung im SceneBuilder stylen	158
7.4.1	Wie kann ich im SceneBuilder eine CSS-Datei registrieren?	159
7.4.2	Wie erzeuge ich einen LCD-Screen-Effekt?	160
7.4.3	Wie kann ich in CSS einen Font laden?	161
7.4.4	Wie erzeuge ich Pseudo-3D-Buttons?	161
7.4.5	Wie weise ich einem Node eine Id zu?	162
7.4.6	Wie trage ich im SceneBuilder eine Styleklasse ein?	163
7.4.7	Stylecheck mit dem CSS-Analyzer	164
7.5	Fazit	165

8	Charts erstellen	167
8.1	Ein Diagramm anzeigen	167
8.2	Aufbau der API	168
8.2.1	Achsen	168
8.2.2	Daten	170
8.3	Welche Diagrammtypen gibt es?	170
8.3.1	LineChart und AreaChart	170
8.3.2	PieChart	173
8.3.3	BarChart	174
8.3.4	ScatterChart	176
8.3.5	BubbleChart	177
8.4	Dynamische Charts	179
9	Nebenläufigkeit und Fortschrittsanzeige	181
9.1	JavaFX und die gefühlte Performance	181
9.2	Die Klassen der javafx.concurrent-API	182
9.2.1	Worker	182
9.2.2	Task	183
9.2.3	Service	185
9.2.4	ScheduledService	187
9.3	Fortschritt anzeigen	189
9.3.1	ProgressBar	189
9.3.2	ProgressIndicator	190
9.3.3	Fortschrittskomponenten anpassen	190
9.4	Workshop: Automatische Updates für den Twitter-Client	192
9.4.1	Wie erzeuge ich einen Refresh Task?	192
9.4.2	Wie kann der Task wiederholt aufgerufen werden?	192
9.4.3	Wie rufe ich den Service periodisch auf?	193
9.4.4	Wie aktualisiere ich das UI?	193
9.5	Fazit	193
10	Grafische Anwendungen mit Shapes gestalten	195
10.1	Den SceneGraph verstehen – Unterschiede zu Swing	195
10.2	Nodes für Grundformen	196
10.2.1	Rectangle	196
10.2.2	Circle	196
10.2.3	Ellipse	197
10.2.4	Schnittmengen bilden	198
10.2.5	Line	198
10.2.6	Eigenschaften des Linienstrichs verändern	198
10.2.7	Polyline und Polygon	199

10.2.8	Arc, CubicCurve und QuadCurve	200
10.2.9	Text	202
10.3	Pfade und SVG	202
10.4	Grafiken mit Effekten versehen	204
10.4.1	Wie erzeuge ich einen Schlagschatten?	205
10.4.2	Wie verwende ich den InnerShadow?	205
10.4.3	Wie verwende ich Reflection?	206
10.4.4	Der Blend-Effekt	206
10.4.5	Der Bloom-Effekt	208
10.4.6	Wie kann ich Effekte verknüpfen?	209
11	Animationen erstellen	211
11.1	Vorgefertigte Transitionen verwenden	211
11.1.1	Eine Transition steuern	212
11.1.2	Start- und Endwert setzen	213
11.1.3	Eine Transition in Schleifen abspielen	213
11.1.4	PathTransition	214
11.2	Animationen in parallelen und sequenziellen Transitionen organisieren	214
11.3	Timeline und KeyFrames	217
11.3.1	Besonderheit des ersten KeyFrame	219
11.3.2	Interpolierbare Typen und das Interface Interpolatable	219
11.4	Interpolatoren	221
11.4.1	Vordefinierte Interpolatoren	221
11.4.2	Einen eigenen Interpolator erstellen	222
11.5	Eigene Transitionen erstellen	223
12	Auf systemnahen Input reagieren	227
12.1	Welche sind die wichtigsten Klassen?	227
12.1.1	Target	227
12.1.2	Source	228
12.1.3	EventType	228
12.1.4	EventHandler	228
12.2	Wie werden Events verteilt?	228
12.2.1	Event Target Selection	229
12.2.2	Event Route Construction	229
12.2.3	Event Capturing Phase	229
12.2.4	Event Bubbling Phase	229
12.3	Wie reagiert man auf Events?	229
12.3.1	EventHandler Properties	230
12.3.2	EventHandler- und EventFilter-Registrierung	230
12.3.3	EventDispatcher	233

13	Die WebView nutzen	235
13.1	Wie zeige ich eine Webseite an?	235
13.2	Wie kann ich JavaScript ausführen?	239
13.2.1	Geht das auch einfacher?	242
13.3	Wie funktionieren Callbacks von JavaScript nach Java?	244
13.4	Workshop: Twitter-Authentifizierung einbauen	245
13.4.1	Erweiterungsmöglichkeiten	248
14	Audio und Video einbinden	249
14.1	Wie kann ich Audio abspielen?	249
14.1.1	Wie überwache ich den Zustand des Players?	252
14.1.2	Wie kontrolliere ich den Sound?	253
14.1.3	Welche Audioformate werden unterstützt?	254
14.2	Wie kann ich eine Videodatei abspielen?	254
14.2.1	Welche Videoformate werden unterstützt?	255
14.3	Wie kann ich damit auch YouTube-Videos abspielen?	255
15	Canvas-API	257
15.1	Retained Mode vs. Immediate Mode	258
15.2	Elemente anzeigen	259
15.2.1	Einfache Formen	259
15.2.2	Komplexe Formen (Path)	260
15.2.3	Den Stil ändern	261
15.2.4	Bilder	263
15.2.5	Text	263
15.2.6	Einzelne Pixel	264
15.3	Fortgeschrittene Techniken	265
15.3.1	Transformationen	265
15.3.2	State Stack	265
15.3.3	Global Alpha und Blend Mode	265
15.3.4	Effekte	266
15.4	Unterschiede zum HTML Canvas	266
15.4.1	Methoden zur Anzeige von Shapes	266
15.4.2	Typen der Properties	267
15.4.3	Hitareas	267
15.4.4	Effekte	268
15.5	Aufgabe	268
15.5.1	Erweiterungsmöglichkeiten	271

16	JavaFX und Internationalisierung	273
16.1	Wie verwendet man Locale und ResourceBundles?	273
16.1.1	Wie verwendet man ResourceBundles mit FXML?	276
	Index	279