

Inhaltsverzeichnis

1	Einleitung & Grundlagen	1
1.1	Was behandeln wir in dem einleitenden Kapitel?	1
1.2	Die Welt von Java und JavaFX	2
1.2.1	Was ist Java?	2
1.2.2	Etwas zur Historie von Java	4
1.2.3	Wo findet Java heutzutage Anwendung?	4
1.2.4	Die Java-Plattform	5
1.3	Was sind JavaFX sowie JavaFX Script und was hat es mit OpenJFX auf sich?	7
1.3.1	Was ist OpenJFX?	9
1.3.2	Die Architektur	10
1.3.3	JavaFX-Oberflächen ohne Java programmieren zu müssen	12
1.4	Was benötigen Sie?	13
1.4.1	Die Hardware und das Betriebssystem	13
1.4.2	Die Java-Basisumgebung	14
1.4.3	Download und Bereitstellung von JavaFX	16
1.4.4	Integrierte Entwicklungsumgebungen für JavaFX	17
1.4.5	Der Scene Builder	19
1.5	Ein erstes JavaFX-Beispiel mit dem Scene Builder	19
1.5.1	Ein erster Blick auf den Scene Builder	19
1.5.2	Bauen wir uns eine FXML-Oberfläche	22
1.6	JavaFX-Programme mit Gradle und Maven	26
1.6.1	Ein Grundprojekt mit Maven	26
1.6.2	Ein Grundprojekt mit Gradle	27
1.7	Ein erstes JavaFX-Programm mit NetBeans	27
1.7.1	Eine echte JavaFX Application erzeugen	28
1.7.2	Build-Tools in NetBeans verwenden	30
1.7.3	Ein Ant-Java-Projekt für JavaFX konfigurieren	30
1.7.4	Ein Blick auf die Code-Strukturen – das Projekt mit Substanz füllen	36

1.8	Eine JavaFX-Applikation mit Eclipse	40
1.8.1	Die JavaFX-Applikation	40
1.8.2	Vorhandene Projekte in NetBeans oder Eclipse importieren	41
1.8.3	Eine JavaFX-Applikation nur mit dem JDK, dem JavaFX SDK und einem Editor	42
1.9	Verteilen und Ausführen von JavaFX-Applikationen	46
1.9.1	Erstellen von JAR-Dateien	46
1.9.2	Das Archiv aus einer IDE erstellen – Eclipse	48
1.9.3	Mit NetBeans die JavaFX-Applikationen weitergeben	49
2	Hilfe und Basiswissen	51
2.1	Was behandeln wir in diesem Kapitel?	51
2.2	Die Dokumentationsseiten von Oracle	51
2.2.1	Die API-Dokumentation	52
2.3	Standardbeispiele	53
2.4	Support durch Ihre IDE	54
2.5	Wichtige grundlegende Regeln für Java	57
2.6	CSS – Style Sheets für das bessere Aussehen	58
2.6.1	Regeln, immer wieder Regeln	59
3	FXML und der JavaFX Scene Builder	61
3.1	Was behandeln wir in diesem Kapitel?	61
3.2	Eine FXML-Applikation mit NetBeans erstellen	61
3.2.1	Eine FXML-Applikation anlegen	62
3.2.2	Die Struktur eines FXML-Projekts	62
3.3	Das Konzept einer grafischen Oberfläche – die Bedeutung von Bäumen und der JavaFX Scene Graph	67
3.3.1	Komponenten	68
3.3.2	Fenster und Container	68
3.4	Container in FXML	68
3.4.1	Das AWT – Großvater GUI	69
3.4.2	Das Konzept der Layoutmanager	70
3.4.3	Swing	74
3.4.4	Das grundsätzliche GUI-Konzept bei JavaFX	74
3.5	Die verschiedenen Container bei JavaFX	75
3.5.1	Absolute Layouts – AnchorPane und Pane	76
3.5.2	Accordion und TitlePane	76
3.5.3	Der BorderPane-Container	77
3.5.4	Fließende Inhalte – FlowPane, HBox und VBox	78
3.5.5	Tabellen beziehungsweise Gitter – der GridPane-Container	80

3.5.6	Stapelverarbeitung mit StackPane	81
3.5.7	Scrollbare Bereiche – ScrollPane	81
3.5.8	Gesplittete Ansichten – SplitPane	82
3.5.9	Tab-Strukturen – TabPane und Tab	83
3.5.10	Ein TilePane	84
3.5.11	Platz da – TitledPane	84
3.5.12	Die Toolbar	85
3.6	Die Controls	85
3.6.1	Beginn eines Praxisprojekts – ein Taschenrechner	87
3.6.2	Ein weiteres Praxisprojekt – ein Bildbetrachter	91
3.7	Grafik und Zeichnen unter JavaFX	93
3.7.1	Allgemeines zum Zeichnen und Malen auf grafischen Java-Oberflächen	94
3.8	Java-2D und dessen Erbe in JavaFX – gutes Aussehen, Effekte, komplexe Formen und Transformationen	96
3.8.1	Wer hat's erfunden?	96
3.8.2	Der Koordinatenraum und Transformationen	96
3.8.3	Füllungen und Rahmen	98
3.8.4	Transparenz	100
3.8.5	Diagramme – Charts	101
4	Gestalten mit dem Inspector des Scene Builders	103
4.1	Was behandeln wir in dem Kapitel?	103
4.2	Grundsätzliche Konfigurationen im Inspector	103
4.3	Das Properties-Register	104
4.3.1	Texteigenschaften	107
4.3.2	Knoten und Extras	107
4.3.3	Effekte	107
4.4	JavaFX CSS zur Gestaltung	109
4.4.1	Selektoren	110
4.4.2	Die spezifischen JavaFX-Bezeichner	110
4.4.3	Das Default Style Sheet	111
4.4.4	Eigene Style Sheets und die Anwendung im Scene Builder	113
4.5	Die Layout-Kategorie im Inspector	116
4.6	Weiterentwicklung der Praxisprojekte	117
4.6.1	Taschenrechner 2.0	117
4.6.2	Bildbetrachter 2.0	118

5	Behind the scene – der Aufbau von FXML	121
5.1	Was behandeln wir in diesem Kapitel?	121
5.2	Warum FXML?	121
5.3	XML-Grundlagen	122
5.3.1	Wo kommt XML her und was macht man damit?	122
5.3.2	Erstellen und Anzeigen von XML-Dokumenten	123
5.3.3	Der Aufbau von XML-Dokumenten	123
5.3.4	Bestandteile eines XML-Dokuments	124
5.3.5	Die Syntax eines XML-Dokuments – Wohlgeformtheit	129
5.4	Die Struktur von FXML	131
5.4.1	FXML-Elemente	131
5.4.2	FXML-Attribute	134
5.4.3	Static Properties – Statische Eigenschaften	137
5.5	Weiterentwicklung des Bildbetrachters	137
6	JavaScript und der JavaFX Scene Builder	141
6.1	Was behandeln wir in diesem Kapitel?	141
6.2	Einige Hintergründe zu JavaScript	141
6.2.1	JavaScript im Web	142
6.2.2	Syntax und Aufbau	142
6.2.3	Funktionen	146
6.2.4	JavaScript und Objekte	147
6.3	Ereignisbehandlung	148
6.3.1	Vorüberlegungen zur Ereignisbehandlung	148
6.3.2	Ereignisse und die Reaktion darauf	149
6.3.3	Das Code-Register im Inspector des Scene Builder	149
6.3.4	Reaktion im Java-Controller	151
6.3.5	Mit Skriptsprachen reagieren	154
6.4	Weiterentwicklung des Taschenrechners	161
7	Java – die Schnellbahn	167
7.1	Was behandeln wir in diesem Kapitel?	167
7.2	Das JavaFX-API	167
7.2.1	Die wichtigsten Pakete	169
7.3	Die grundsätzliche Java-Struktur einer JavaFX-Applikation	169
7.3.1	Das Hauptprogramm – ein Objekt vom Typ javafx.application.Application	171
7.4	Die Layout-Panes	173
7.4.1	Elemente in Panes	174
7.5	Die Größe und Position von Elementen	175
7.6	Style Sheets in Java verwenden	176
7.6.1	Vorgabeklassen und keine Vorgabeklassen	176

7.6.2	Zuordnung von Style Sheet-Klassen	177
7.7	Die JavaFX UI Controls	178
7.7.1	Ein praktisches Beispiel für eine GUI	178
7.8	Funktionalität hinzufügen – Überlegungen und Hintergrundinformationen zum Eventhandling	180
7.8.1	Aktions-Setter	181
7.8.2	Die EventHandler-Schnittstelle und die handle()-Methode	181
7.8.3	Dem Beispiel Funktionalität hinzufügen	183
7.8.4	Eventhandling nur mit Java	186
7.8.5	Einen Event-Controller verwenden	188
7.8.6	Weiterentwicklung des Bildbetrachters	189
7.9	Umgang mit Canvas-Elementen	195
7.9.1	Der Grafikkontext	196
7.9.2	Das Koordinatensystem und die möglichen Ausgabemethoden	196
8	Ereignisse und mehr	201
8.1	Was behandeln wir in diesem Kapitel?	201
8.2	Ereignisbehandlung	201
8.2.1	Die JavaFX- Events	202
8.2.2	Noch tiefere Hintergründe – vom Blubbern, Binden, Callbacks und Triggern	202
8.2.3	Die Auswahl des Ziels	204
8.2.4	Das Auffangen oder Filtern des Ereignisses	205
8.3	Konkrete Reaktionen auf Ereignisse	206
8.3.1	EventHandler	207
8.3.2	Einige praktische Beispiele	212
8.3.3	Eventfilter	218
8.4	Die Controllerklasse bei FXML	220
8.4.1	Details zur Controllerklasse	222
8.5	JavaFX Properties und Binding	226
8.5.1	Abhängigkeiten und das API	226
8.5.2	Definition und Namenskonventionen bei Properties	227
8.5.3	Listener bei Properties	229
8.5.4	Anwendung des High-Level Binding API	232
9	JavaFX und Swing	239
9.1	Was behandeln wir in diesem Kapitel?	239
9.2	JavaFX in Swing	239
9.2.1	Synchronisation und fremde Welten	240
9.2.2	Die Methoden runLater() und invokeLater()	241
9.2.3	Ein Beispiel für die Integration von JavaFX in Swing	241

9.3	Swing in JavaFX	248
10	HTML-Content	251
10.1	Was behandeln wir in diesem Kapitel?	251
10.2	HTML-Content in JavaFX-Applikationen	251
10.2.1	Das eingebettete Browser-API	251
10.3	Hyperlinks und Adresszeilen	254
11	Diagramme	259
11.1	Was behandeln wir in diesem Kapitel?	259
11.2	Grundlagen	259
11.3	Einführung in JavaFX Charts	260
11.3.1	Die verfügbaren Typen	260
11.3.2	Die Datenbasis	261
11.4	Diagramme ohne Achsen	261
11.4.1	Ein praktisches Beispiel für ein Kuchendiagramm mit Java	261
11.4.2	Konfiguration und Einstellungen von Diagrammen	263
11.4.3	Das praktische Beispiel für ein Kuchendiagramm mit FXML	265
11.5	Diagramme mit Achsen	267
11.5.1	Datenserien	268
11.5.2	Ein praktisches Beispiel für ein Diagramm mit Achsen: ein Balkendiagramm	268
11.6	Dynamik und Ereignisbehandlung in Diagrammen	272
11.6.1	Dynamisches Kuchendiagramm – Teil 1	273
11.6.2	Datenpunkte aktualisieren – die Methode set()	277
11.6.3	Ereignisverarbeitung für Diagrammelemente	286
11.6.4	Ein Beispiel für ein Kuchendiagramm mit der Behandlung von Ereignissen	286
12	Spezialitäten	293
12.1	Was behandeln wir in diesem Kapitel?	293
12.2	JavaFX Collections	293
12.2.1	Grundlagen zu dynamischen Datenstrukturen	294
12.2.2	Details zu JavaFX Collections	297
12.2.3	Beispiel 1 – eine dynamische Liste zur Erweiterung von Inhalten in einem mehrzeiligen Textfeld	298
12.2.4	Beispiel 2 – dynamisch Steuerelemente hinzufügen	302
12.2.5	Beispiel 3 – Verwenden einer ObservableMap	305
12.2.6	Beispiel 4 – die Klassenmethoden von FXCollections nutzen	309

12.3	Multithreading	312
12.3.1	Threads laufen lassen – run() und start()	312
12.3.2	Threads abbrechen	313
12.3.3	Multithreading in JavaFX – javafx.concurrent	313
12.4	Drag & Drop	333
12.4.1	Hintergrundinformationen	334
12.4.2	Ein einfaches Drag & Drop-Beispiel	334
13	OOP- und Java-Crashkurs	341
13.1	Syntaxfragen	341
13.1.1	Datentypen	341
13.2	Java und die OOP	342
13.2.1	Kernkonzepte der OOP	342
13.2.2	Objekte und Klassen	343
13.2.3	Identifizieren Sie sich – Botschaften	345
13.2.4	Klassen	346
13.3	Variablen und Eigenschaften	347
13.3.1	Deklaration von Variablen und Eigenschaften	348
13.3.2	Die Methodendeklaration	348
13.3.3	Konstruktoren und Destruktoren – das Speichermanagement	350
13.4	Lokale und anonyme Klassen	351
13.4.1	Klassendeklarationen in Methoden	352
13.4.2	Anonyme Klassen	352
13.5	Pakete und die import-Anweisung	353
13.5.1	Die Zuordnung einer Klasse zu einem Paket und das Default-Paket	354
13.5.2	Die Suche nach Paketen	355
13.5.3	Die import-Anweisung	355
13.6	Vererbung	356
13.6.1	Superklasse und Subklasse	357
13.6.2	Die technische Umsetzung einer Vererbung	358
13.7	Überschreiben und Überladen	359
13.8	Information Hiding und Zugriffsschutz	359
13.8.1	Indirekte Zugriffe über Getter und Setter	360
13.9	Abstrakte Klassen und Schnittstellen	360
13.9.1	Was ist eine abstrakte Klasse?	361
13.9.2	Was ist eine Schnittstelle?	361
	Stichwortverzeichnis	363