

Inhaltsverzeichnis

| | |
|---|-----------|
| Vorwort | 11 |
| Geleitwort | 17 |
| 1 Einführung | 19 |
| 1.1 Was ist Eclipse? | 19 |
| 1.1.1 Eclipse ist ein IDE-Framework | 20 |
| 1.1.2 Eclipse ist eine Tools Platform | 20 |
| 1.1.3 Eclipse ist eine Anwendungsplattform | 21 |
| 1.1.4 Eclipse ist eine Open Source Community | 23 |
| 1.1.5 Die Eclipse Public License und die Eclipse Foundation | 24 |
| 1.2 Weichenstellung für die Zukunft | 25 |
| 1.2.1 Von e4 zu Eclipse 4.2 | 26 |
| 1.2.2 Einfacher und moderner programmieren | 27 |
| 1.3 Warum soll ich e4 erlernen? | 28 |
| 1.4 Aus welchen Komponenten besteht das Eclipse 4.2 SDK? | 29 |
| 1.4.1 Eclipse 4 Application Platform | 29 |
| 1.4.2 Compatibility Layer | 30 |
| 1.4.3 Platform, JDT, PDE | 31 |
| 1.5 Aus welchen Komponenten besteht die Eclipse 4 Application Platform? | 31 |
| 1.5.1 Dependency Injection | 32 |
| 1.5.2 IEclipseContext | 32 |
| 1.5.3 Core Services | 33 |
| 1.5.4 Application Model | 33 |
| 1.5.5 Rendering Engine | 33 |
| 1.5.6 Declarative Styling | 34 |

| | |
|---|-----------|
| 2 Erste Schritte | 35 |
| 2.1 Eclipse 4.2 SDK einrichten | 35 |
| 2.2 Anwendungsgerüst erstellen | 36 |
| 2.2.1 Plugin-Projekt einrichten | 36 |
| 2.2.2 Application Model erzeugen | 38 |
| 2.2.3 Das Modell mit der Plattform zusammenbringen | 39 |
| 2.2.4 Die Anwendung starten | 42 |
| 2.3 Die Anwendung fertigstellen | 43 |
| 2.3.1 Früher Views, heute Parts | 43 |
| 2.3.2 Views testen und wiederverwenden | 45 |
| 2.3.3 Part ins Application Model aufnehmen | 46 |
| 2.4 Die Anwendung um einen Handler erweitern | 47 |
| 2.4.1 Toolbar und Handler deklarativ hinzufügen | 47 |
| 3 Entwicklungsumgebung | 55 |
| 3.1 e4-Tools | 55 |
| 3.1.1 e4-Tools installieren | 55 |
| 3.1.2 Überblick | 57 |
| 3.2 Wizards | 58 |
| 3.2.1 e4 Application Project Wizard | 58 |
| 3.2.2 Classes | 61 |
| 3.2.3 Model | 63 |
| 3.3 Application Model Editor | 65 |
| 3.3.1 Die Darstellung im Application Model Editor | 66 |
| 3.3.2 Änderungen am Modell vornehmen | 68 |
| 3.4 Live Editor | 70 |
| 3.4.1 Das Eclipse SDK mit dem Live Editor betrachten | 70 |
| 3.4.2 Den Live Editor in eigenen Programmen verwenden | 70 |
| 3.5 CSS-Tooling | 77 |
| 3.5.1 CSS-Editor | 77 |
| 3.5.2 CSS Spy | 77 |
| 3.5.3 CSS Scratchpad | 80 |
| 3.6 Window Builder | 81 |
| 3.6.1 Installation | 81 |
| 3.6.2 Window Builder und e4 | 83 |
| 3.6.3 Die Zukunft von Window Builder | 85 |

| | |
|---|------------|
| 4 Application Model | 87 |
| 4.1 Einführung | 87 |
| 4.1.1 Organisation klassischer Eclipse 3.x-Anwendungen | 87 |
| 4.1.2 Organisation von Eclipse 4.x-Anwendungen | 89 |
| 4.1.3 Aufbau des Application Models | 90 |
| 4.1.4 Modell-Elemente mit externen Ressourcen verbinden | 91 |
| 4.1.5 Eclipse Modelling Framework | 93 |
| 4.2 Visuelle Elemente im Application Model | 93 |
| 4.2.1 Windows | 94 |
| 4.2.2 Parts | 98 |
| 4.2.3 Parts am Beispiel eines kleinen Texteditors | 106 |
| 4.2.4 PartDescriptoren im Einsatz | 112 |
| 4.2.5 Perspektiven | 116 |
| 4.2.6 Menüs | 126 |
| 4.2.7 Window Trims und ToolBars | 131 |
| 4.3 Nicht visuelle Elemente | 137 |
| 4.3.1 Handler | 138 |
| 4.3.2 Commands | 139 |
| 4.4 Programmatischer Zugriff | 140 |
| 4.5 Das Application Model erweitern | 143 |
| 4.5.1 Model Fragment | 145 |
| 4.5.2 Model Processor | 147 |
| 5 Das Programmiermodell | 149 |
| 5.1 Einführung | 149 |
| 5.2 Der Eclipse Context | 151 |
| 5.2.1 @Named | 153 |
| 5.2.2 @Optional | 154 |
| 5.2.3 @Active | 154 |
| 5.3 Objekte injizieren | 154 |
| 5.3.1 Konstruktoren | 155 |
| 5.3.2 Felder | 155 |
| 5.3.3 Methoden | 155 |

| | |
|--|------------|
| 5.4 Annotationen | 156 |
| 5.4.1 @PostConstruct und @PreDestroy | 156 |
| 5.4.2 @Focus | 157 |
| 5.4.3 @Persist | 157 |
| 5.4.4 @PersistState | 158 |
| 5.4.5 @Execute und @CanExecute | 158 |
| 5.4.6 Lifecycle-Annotationen | 158 |
| 5.5 Den Context erweitern | 159 |
| 5.5.1 @Creatable | 159 |
| 5.5.2 Objekte manuell in den Context einfügen | 160 |
| 5.6 Die Injektion manuell durchführen | 162 |
| 5.7 Übersicht der Annotationen | 164 |
| 6 Services | 165 |
| 6.1 Selection Service | 166 |
| 6.2 Preferences | 167 |
| 6.3 Model Service und Part Service | 168 |
| 6.4 Event Broker | 169 |
| 6.4.1 Objekte an den Event Broker senden | 170 |
| 6.4.2 Vom Event Broker benachrichtigt werden | 171 |
| 6.5 Translation Service | 173 |
| 6.6 Eigene OSGi Services definieren | 174 |
| 6.6.1 OSGi Declarative Services | 177 |
| 7 CSS-Styling | 181 |
| 7.1 CSS und Eclipse 4.2 | 181 |
| 7.1.1 Das Aussehen der Eclipse 4 IDE mit CSS verändern | 183 |
| 7.1.2 CSS in Eclipse-4-Anwendungen verwenden | 186 |
| 7.1.3 Farben und Gradienten | 192 |
| 7.2 CSS-Selektoren | 195 |
| 7.2.1 CSS-Selektoren nach Typ | 195 |
| 7.2.2 CSS-Selektoren nach ID | 195 |
| 7.2.3 CSS-Selektoren nach Klasse | 196 |
| 7.2.4 Forgeschrittene Selektoren | 197 |

| | |
|--|------------|
| 8 Build und Deployment | 205 |
| 8.1 Build-Werkzeuge | 206 |
| 8.1.1 Apache Ant | 206 |
| 8.1.2 Apache Maven | 206 |
| 8.2 Tycho | 207 |
| 8.2.1 Tycho bringt OSGi-Metadaten mit dem POM zusammen | 208 |
| 8.2.2 Tycho installieren und einrichten | 209 |
| 8.3 Packaging Types | 211 |
| 8.3.1 eclipse-plugin | 212 |
| 8.3.2 eclipse-test-plugin | 217 |
| 8.3.3 eclipse-feature | 228 |
| 8.3.4 eclipse-repository | 231 |
| 8.3.5 Produkte bauen | 236 |
| 9 Migration und Ausblick | 241 |
| 9.1 Migration | 241 |
| 9.1.1 Der Compatibility Layer | 241 |
| 9.1.2 Migration von 3.x? | 242 |
| 9.1.3 Reiner Compatibility Layer | 242 |
| 9.1.4 Eine reine Eclipse-4-Applikation | 243 |
| 9.1.5 Eine Eclipse-4-Applikation mit 3.x Komponenten | 243 |
| 9.1.6 Compatibility Layer und Eclipse 4 Plugins | 243 |
| 9.1.7 Eclipse 4 kompatibel entwickeln | 245 |
| 9.2 Ausblick | 246 |
| 9.2.1 Eclipse 3.x | 246 |
| 9.2.2 Eclipse 4.2.x and beyond | 247 |
| Autoreninfo | 249 |
| Stichwortverzeichnis | 251 |