

# Contents

## Foreword — XI

- 1 Mastering App Development Requires More than Coding Skills — XII
- 2 Why is JavaScript a Good Choice for Building Web Apps? — XII
- 3 Why is Java a Good Choice for Building Web Apps? — XIII
- 4 Comparing Java with JavaScript — XIII
- 5 Good Programs Are: Effective, Robust, Efficient and Maintainable — XIV
- 6 Run the Apps and Get Their Code — XIV

## List of Figures — XV

## List of Tables — XVII

## Part I Getting Started

### 1 A Quick Tour of the Foundations of Web Apps — 3

- 1.1 The World Wide Web (WWW) — 3
- 1.2 HTML and XML — 3
- 1.3 Styling Web Documents and User Interfaces with CSS — 9
- 1.4 JavaScript – “the assembly language of the Web” — 10
- 1.5 Accessibility for Web Apps — 11
- 1.6 Quiz Questions — 12

### 2 More on JavaScript — 14

- 2.1 JavaScript Basics — 14
- 2.2 Storing Database Tables with JavaScript’s localStorage API — 35

### 3 Building a Minimal Web App with Plain JS in Seven Steps — 37

- 3.1 Step 1 – Set up the Folder Structure — 38
- 3.2 Step 2 – Write the Model Code — 39
- 3.3 Step 3 – Initialize the Application — 44
- 3.4 Step 4 – Implement the *Create* Use Case — 44
- 3.5 Step 5 – Implement the *Retrieve/List All* Use Case — 46
- 3.6 Step 6 – Implement the *Update* Use Case — 47
- 3.7 Step 7 – Implement the *Delete* Use Case — 49
- 3.8 Run the App and Get the Code — 50
- 3.9 Possible Variations and Extensions — 50
- 3.10 Points of Attention — 52
- 3.11 Practice Projects — 55

|          |                                                                    |
|----------|--------------------------------------------------------------------|
| <b>4</b> | <b>Building a Minimal Web App with Java EE in Seven Steps — 58</b> |
| 4.1      | Java Basics — 59                                                   |
| 4.2      | Step 1 – Set up the Folder Structure — 61                          |
| 4.3      | Step 2 – Write the Model Code — 63                                 |
| 4.4      | Step 3 – Configure the App — 68                                    |
| 4.5      | Step 4 – Implement the <i>Create</i> Use Case — 75                 |
| 4.6      | Step 5 – Implement the <i>Retrieve/List All</i> Use Case — 77      |
| 4.7      | Step 6 – Implement the <i>Update</i> Use Case — 79                 |
| 4.8      | Step 7 – Implement the <i>Delete</i> Use Case — 82                 |
| 4.9      | Style the User Interface with CSS — 83                             |
| 4.10     | Run the App and Get the Code — 84                                  |
| 4.11     | Possible Variations and Extensions — 85                            |
| 4.12     | Points of Attention — 86                                           |
| 4.13     | Practice Projects — 87                                             |
| 4.14     | Quiz Questions — 88                                                |
| <b>5</b> | <b>Information Modeling — 91</b>                                   |
| 5.1      | Classes with Properties and Methods — 92                           |
| 5.2      | Connecting Classes with Associations — 95                          |
| 5.3      | From a Conceptual Model via a Design Model to Class Models — 95    |
| 5.4      | Excursion: Formalizing Information Models with RDF and OWL — 97    |
| 5.5      | Summary — 105                                                      |
| 5.6      | Exercises — 106                                                    |
| <b>6</b> | <b>Application Architecture — 107</b>                              |
| 6.1      | The Model-View-Controller (MVC) Architecture Metaphor — 108        |
| 6.2      | The Onion Architecture Metaphor — 110                              |
| 6.3      | “Logical” versus “Physical” User Interface — 111                   |
| 6.4      | MVC Web Applications — 112                                         |
| 6.5      | Deployment Architectures — 113                                     |

## **Part II Constraint Validation**

|          |                                                                    |
|----------|--------------------------------------------------------------------|
| <b>7</b> | <b>Integrity Constraints and Data Validation — 117</b>             |
| 7.1      | Introduction — 117                                                 |
| 7.2      | Integrity Constraints — 118                                        |
| 7.3      | Responsive Validation — 129                                        |
| 7.4      | Constraint Validation in MVC Applications — 130                    |
| 7.5      | Adding Constraints to a Design Model — 131                         |
| 7.6      | Summary — 132                                                      |
| 7.7      | Criteria for Evaluating the Validation Support of Frameworks — 133 |
| 7.8      | Quiz Questions — 133                                               |

- 8 Implementing Constraint Validation in a Plain JS Web App — 136**
  - 8.1 Introduction — 136
  - 8.2 New Issues — 136
  - 8.3 Make a JavaScript Class Model — 137
  - 8.4 Set up the Folder Structure Adding Some Library Files — 139
  - 8.5 Write the Model Code — 140
  - 8.6 Write the View Code — 146
  - 8.7 Run the App and Get the Code — 150
  - 8.8 Possible Variations and Extensions — 151
  - 8.9 Points of Attention — 152
  - 8.10 Practice Projects — 153
  - 8.11 Quiz Questions — 156
- 9 Implementing Constraint Validation in a Java EE Web App — 159**
  - 9.1 Java Annotations for Persistent Data Management and Constraint Validation — 159
  - 9.2 New Issues — 162
  - 9.3 Make an Entity Class Model — 163
  - 9.4 Write the Model Code — 164
  - 9.5 Write the View Code — 168
  - 9.6 Defining a Custom Validation Annotation — 171
  - 9.7 Run the App and Get the Code — 173
  - 9.8 Possible Variations and Extensions — 173
  - 9.9 Practice Projects — 177
  - 9.10 Quiz Questions — 179

## **Part III Enumerations**

- 10 Enumerations and Enumeration Attributes — 183**
  - 10.1 Enumerations — 183
  - 10.2 Enumeration Attributes — 187
  - 10.3 Enumerations in Computational Languages — 188
  - 10.4 Dealing with Enumeration Attributes in a Design Model — 191
  - 10.5 Quiz Questions — 192
- 11 Implementing Enumeration Attributes in a Plain JS Web App — 194**
  - 11.1 New Issues — 194
  - 11.2 Make a JavaScript Class Model — 195
  - 11.3 Add the Library File `Enumeration.js` — 196
  - 11.4 The Meta-Class `Enumeration` — 196
  - 11.5 Write the Model Code — 198
  - 11.6 Write the View Code — 203

- 11.7 Run the App and Get the Code — 207
- 11.8 Practice Projects — 207
- 11.9 Quiz Questions — 210
- 12 Implementing Enumeration Attributes in a Java EE Web App — 211**
  - 12.1 New Issues — 211
  - 12.2 Make an Entity Class Model — 212
  - 12.3 Write the Model Code — 212
  - 12.4 Write the View Code — 218
  - 12.5 Displaying Value Sets for Multi-Valued Enumeration Attributes — 221
  - 12.6 Run the App and Get the Code — 222
  - 12.7 Practice Projects — 222
  - 12.8 Quiz Questions — 224

## **Part IV Special Datatypes and Derived Properties**

- 13 Special Datatypes — 229**
  - 13.1 Boolean Values — 230
  - 13.2 String Patterns — 230
  - 13.3 Special Numeric Data Types — 232
  - 13.4 Calendar Dates and Times — 234
  - 13.5 Quantities — 239
  - 13.6 Complex Data Types — 241
- 14 Derived Properties — 243**
  - 14.1 Virtual Derived Properties — 243
  - 14.2 Materialized Derived Properties — 244
  - 14.3 Dealing with Derived Properties in the User Interface — 245

**Glossary — 246**

**Index — 252**