Contents

Foreword —— VII

List of Figures —— XIII

List of Tables —— XV

Part I: **Associations**

1 Reference Properties and Unidirectional Associations —— 3
- 1.1 References and Reference Properties —— 4
- 1.2 Referential Integrity —— 6
- 1.3 Modeling Reference Properties as Unidirectional Associations —— 6
- 1.4 Representing Unidirectional Associations as Reference Properties —— 8
- 1.5 Adding Directionality to a Non-Directed Association —— 9
- 1.6 Our Running Example —— 10
- 1.7 Eliminating Unidirectional Associations —— 11
- 1.8 Rendering Reference Properties in the User Interface —— 15
- 1.9 Collection Types for Multi-Valued Reference Properties —— 15
- 1.10 Quiz Questions —— 16

2 Implementing Unidirectional Functional Associations with Plain JS —— 19
- 2.1 New JavaScript Elements —— 19
- 2.2 Implementing Single-Valued Reference Properties —— 22
- 2.3 Make a JS Class Model —— 24
- 2.4 New Issues —— 26
- 2.5 Code the Model —— 26
- 2.6 Code the View and Controller —— 33
- 2.7 Quiz Questions —— 36

3 Implementing Unidirectional Non-Functional Associations with Plain JS —— 38
- 3.1 Implementing Multi-Valued Reference Properties —— 38
- 3.2 Make a JS Class Model —— 39
- 3.3 New Issues —— 40
- 3.4 Code the Model —— 41
- 3.5 Code the View —— 45
- 3.6 How to Run the App and Get the Code —— 51
- 3.7 Points of Attention —— 52

Bibliografische Informationen
http://d-nb.info/1127761889

digitalisiert durch DEUTSCHE NATIONAL BIBLIOTHEK

4 Implementing Unidirectional Functional Associations with Java EE — 53
4.1 Implementing Single-Valued Reference Properties in Java — **53**
4.2 Make a Java Entity Class Model — **54**
4.3 New Issues — **55**
4.4 Write the Model Code — **55**
4.5 The View and Controller Layers — **59**

5 Unidirectional Non-Functional Associations with Java EE — 67
5.1 Implementing Multi-Valued Reference Properties in Java — **67**
5.2 Make a Java Entity Class Model — **68**
5.3 New Issues — **69**
5.4 Write the Model Code — **69**
5.5 Write the User Interface Code — **74**
5.6 Run the App and Get the Code — **76**
5.7 Sets Versus Ordered Sets as the Values of Reference Properties — **77**

6 Unidirectional Associations Practice Projects — 79
6.1 Assign a Director and Actors to a Movie — **79**
6.2 Assign Cities and a Capital to a Country, and Members to an Organization — **80**

7 Bidirectional Associations — 83
7.1 Inverse Reference Properties — **83**
7.2 Making an OO Class Model — **86**

8 Implementing Bidirectional Associations with Plain JS — 91
8.1 Make a JavaScript Class Model — **91**
8.2 Write the Model Code — **92**
8.3 Exploit Inverse Reference Properties in the User Interface — **98**

9 Implementing Bidirectional Associations with Java EE — 100
9.1 Make a Java Entity Class Model — **100**
9.2 Write the Model Code — **102**
9.3 Exploiting Derived Inverse Reference Properties in the User Interface — **110**
9.4 Run the App and Get the Code — **112**

10 Special Topics on Associations — 113
10.1 Part-Whole Associations — **113**
10.2 Association Classes — **116**

11 Bidirectional Associations Practice Projects — 118
11.1 Assign Directors/Actors to Movies and Movies to Directors/Actors — **118**
11.2 Assign Cities to a Country and Members to an Organization, and Vice Versa — **119**

Part II: Inheritance in Class Hierarchies

12 Subtyping and Inheritance — 123
12.1 Introducing Subtypes by Specialization — **123**
12.2 Introducing Supertypes by Generalization — **124**
12.3 Intension Versus Extension — **125**
12.4 Type Hierarchies — **126**
12.5 Kinds and Roles — **128**
12.6 The *Class Hierarchy Merge* Design Pattern — **128**
12.7 Subtyping and Inheritance in Computational Languages — **130**
12.8 Quiz Questions — **141**

13 Subtyping with Plain JS — 143
13.1 Subtyping with Constructor-Based Classes — **143**
13.2 Case Study 1: Eliminating a Class Hierarchy — **145**
13.3 Case Study 2: Implementing a Class Hierarchy — **154**

14 Subtyping with Java EE — 160
14.1 Subtyping in Java — **160**
14.2 Case Study 1: Implement a Class Hierarchy with *Single Table Inheritance* — **161**
14.3 Case Study 2: Implement a Class Hierarchy with *Joined Table Inheritance* — **169**
14.4 Run the App — **173**

15 Subtyping Practice Projects — 175
15.1 Project 1 – Person Types and Movie Types — **175**

Glossary — 177

Index — 183