JavaScript Front-End Web App Tutorial Part 6: Inheritance in Class Hierarchies

An advanced tutorial about developing front-end web applications with class hierarchies, using plain JavaScript

Gerd Wagner <G.Wagner@b-tu.de>

JavaScript Front-End Web App Tutorial Part 6: Inheritance in Class Hierarchies: An advanced tutorial about developing front-end web applications with class hierarchies, using plain JavaScript

by Gerd Wagner

Warning: This tutorial manuscript may still contain errors and may still be incomplete in certain respects. Please report any issue to Gerd Wagner at G.Wagner@b-tu.de.

This tutorial is also available in the following formats: PDF [subtyping-tutorial.pdf]. You may run the example app [SubtypingApp/index.html] from our server, or download it as a ZIP archive file [SubtypingApp.zip]. See also our Web Engineering project page [http://web-engineering.info/].

Publication date 2015-11-12

Table of Contents

Foreword v
1. Subtyping and Inheritance 1
    1. Introducing Subtypes by Specialization 1
    2. Introducing Supertypes by Generalization 2
    3. Intension versus Extension 3
    4. Type Hierarchies 4
    5. The *Class Hierarchy Merge* Design Pattern 4
    6. Subtyping and Inheritance in Computational Languages 6
        6.1. Subtyping and Inheritance with OOP Classes 6
        6.2. Subtyping and Inheritance with Database Tables 6
2. Subtyping in a Plain JavaScript Front-End App 10
    1. Subtyping with Constructor-Based Classes 11
    2. Case Study 1: Eliminating a Class Hierarchy 11
        2.1. Make the JavaScript data model 12
        2.2. New issues 13
        2.3. Encode the model classes of the JavaScript data model 14
        2.4. Write the View and Controller Code 16
    3. Case Study 2: Implementing a Class Hierarchy 18
        3.1. Make the JavaScript data model 18
        3.2. Make the entity table model 19
        3.3. New issues 20
        3.4. Encode the model classes of the JavaScript data model 21
    4. Practice Project 22

List of Figures

1.1. The object type `Book` is specialized by two subtypes: `TextBook` and `Biography` 1
1.2. The object types `Employee` and `Author` share several attributes 2
1.3. The object types `Employee` and `Author` have been generalized by adding the common supertype `Person` 2
1.4. The complete class model containing two inheritance hierarchies 3
1.5. A class hierarchy having the root class `Vehicle` 4
1.6. A multiple inheritance hierarchy 4
1.7. The design model resulting from applying the Class Hierarchy Merge design pattern 5
1.8. A class model with a `Person` roles hierarchy 7
1.9. An SQL table model with a single table representing the `Book` class hierarchy 8
1.10. An SQL table model with a single table representing the `Person` roles hierarchy 8
1.11. An SQL table model with the table `Person` as the root of a table hierarchy 9
2.1. The object type `Book` as the root of a disjoint segmentation 10
2.2. The `Person` roles hierarchy 10
2.3. `Student` is a subclass of `Person` 11
2.4. The simplified information design model obtained by applying the Class Hierarchy Merge design pattern 12
2.5. The JavaScript data model 13
2.6. The JavaScript data model of the Person class hierarchy 19
2.7. The entity table model of the Person class hierarchy 20
2.8. Two class hierarchies: `Movie` with two disjoint subtypes and `Person` with two overlapping subtypes. 23

Foreword

This tutorial is Part 6 of our series of six tutorials [http://web-engineering.info/JsFrontendApp] about model-based development of front-end web applications with plain JavaScript. It shows how to build a web app that manages subtype (inheritance) relationships between object types.

The app supports the four standard data management operations (**C**reate/**R**ead/**U**pdate/**D**elete). It is based on the example used in the other parts, with the object types `Book`, `Person`, `Author`, `Employee` and `Manager`. The other parts are:

- Part 1 [minimal-tutorial.html]: Building a **minimal** app.
- Part 2 [validation-tutorial.html]: Handling **constraint validation**.
- Part 3 [enumeration-tutorial.html]: Dealing with **enumerations**.
- Part 4 [unidirectional-association-tutorial.html]: Managing **unidirectional associations**, such as the associations between books and publishers, assigning a publisher to a book, and between books and authors, assigning authors to a book.
- Part 5 [bidirectional-association-tutorial.html]: Managing **bidirectional associations**, such as the associations between books and publishers and between books and authors, also assigning books to authors and to publishers.

You may also want to take a look at our open access book Building Front-End Web Apps with Plain JavaScript [http://web-engineering.info/JsFrontendApp-Book], which includes all parts of the tutorial in one document, dealing with multiple object types ("books", "publishers" and "authors") and taking care of constraint validation, associations and subtypes/inheritance.