How to find and fix N+1 Select Issues with Hibernate

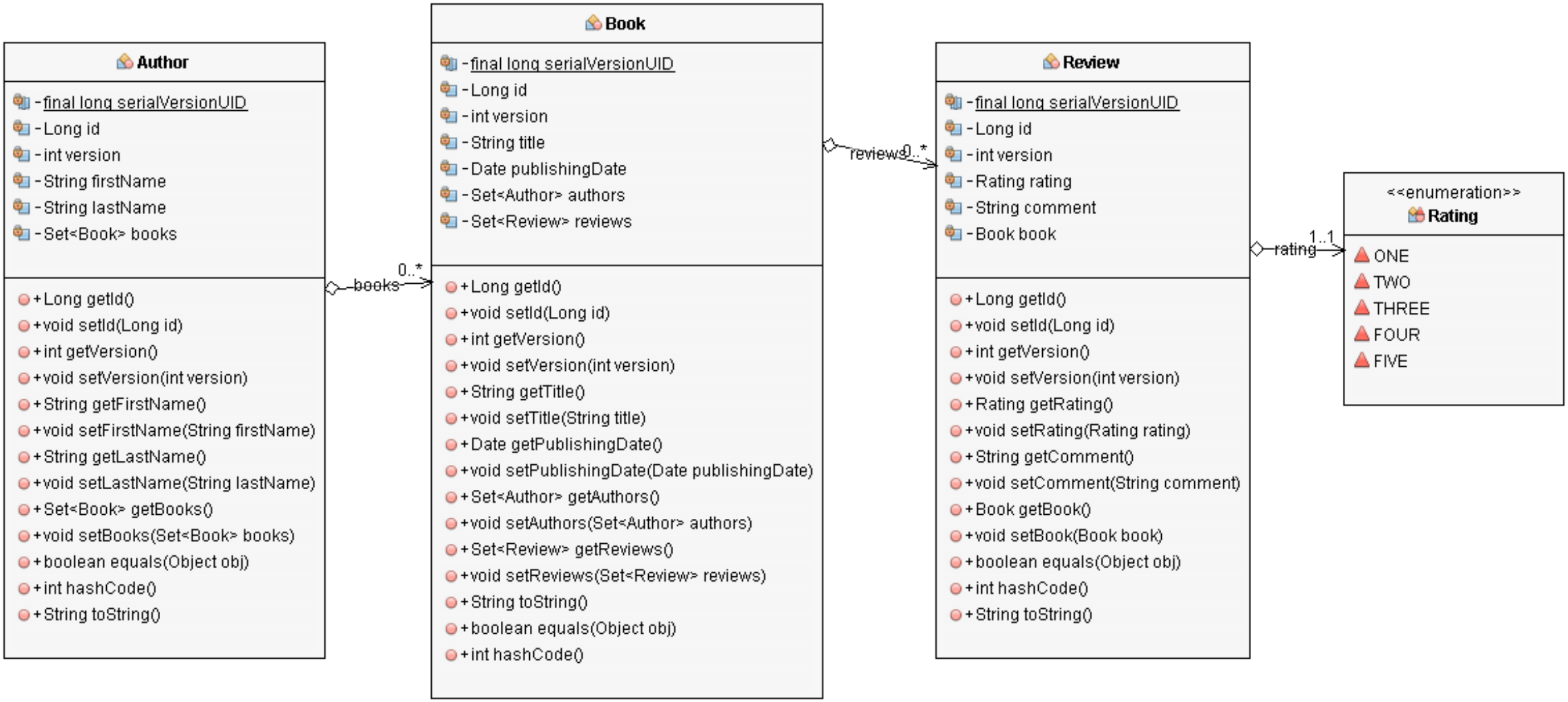
Part 3:

Solving n+1 select issues
with dynamic EntityGraphs

Course Structure

- Part 1:
  - What is the n+1 select issue?
  - How to find it in your project?
- Part 2:
  - Solving n+1 select issues with @NamedEntityGraphs
- Part 3:
  - Solving n+1 select issues with dynamic EntityGraphs

Application



EntityGraph

EntityGraph

- Introduced in JPA 2.1
- Dynamic version of @NamedEntityGraph
- Definition via Java API
- Graph is query independent

EntityGraph

- Define a multi-level EntityGraph and provide it as a hint

```
// define the EntityGraph
EntityGraph graph = this.em.createEntityGraph(Author.class);

Subgraph<Book> bookSubGraph = graph.addSubgraph(Author_.books);

bookSubGraph.addSubgraph(Book_.reviews);


// provide EntityGraph as a hint
this.em.createQuery("SELECT DISTINCT a FROM Author a")
                .setHint("javax.persistence.loadgraph", graph);
```

EntityGraph

- Fetch graph
  - Eager loading for all elements of the graph
  - Lazy loading for all other attributes
- Load graph
  - Eager loading for all elements of the graph
  - Loads all other attributes with their defined FetchType
- **Hibernate always uses a load graph**
  - HHH-8776

EntityGraph

- Advantages
  - Query specific EAGER loading
  - Definition of the graph is independent of the query
  - Dynamic creation at runtime
- Disadvantages
  - Creates cartesian product

Want to learn how to identify and fix other Hibernate performance issues?

Join my

Hibernate Performance Tuning Online Training:

www.thoughts-on-java.org/course-hibernate-performance-tuning