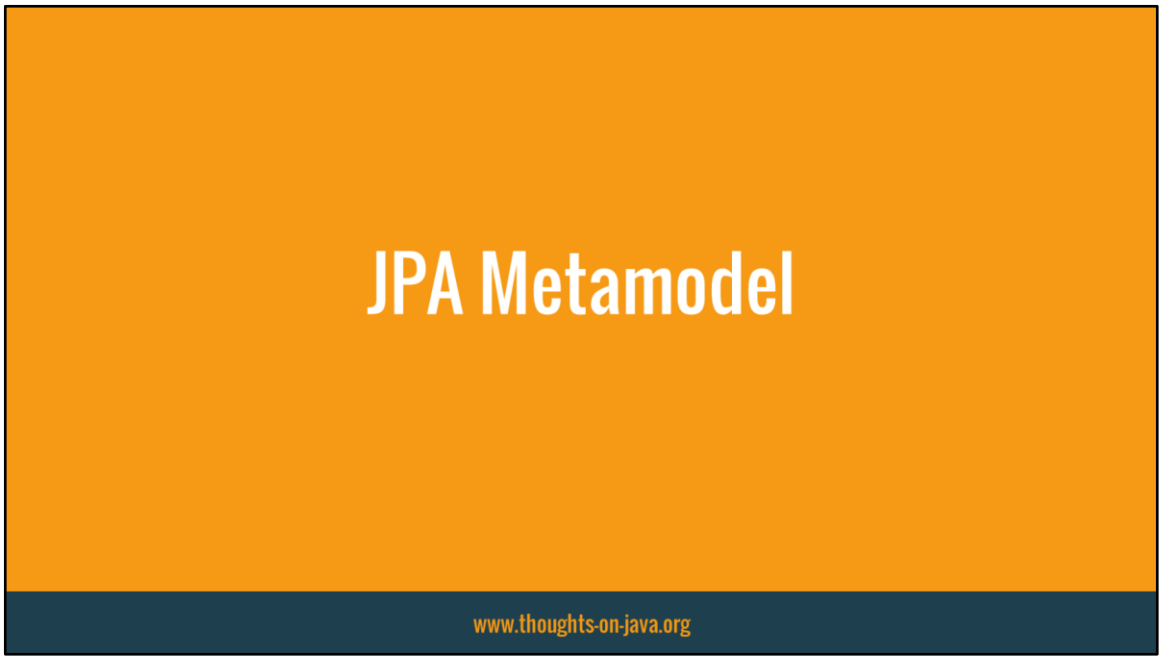
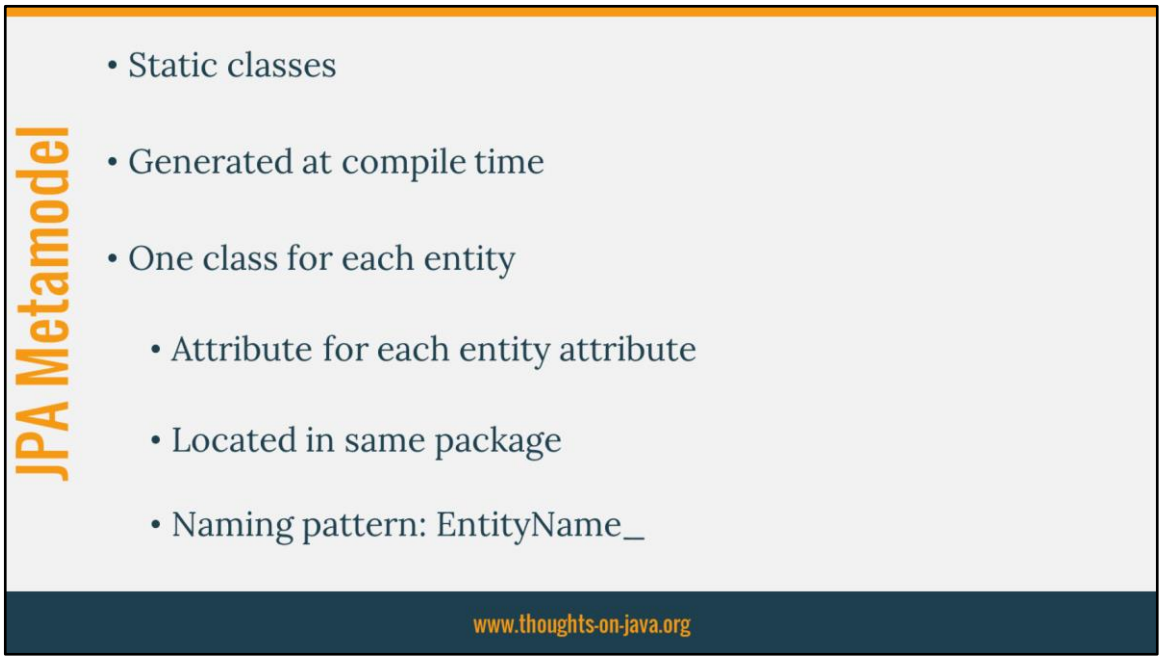
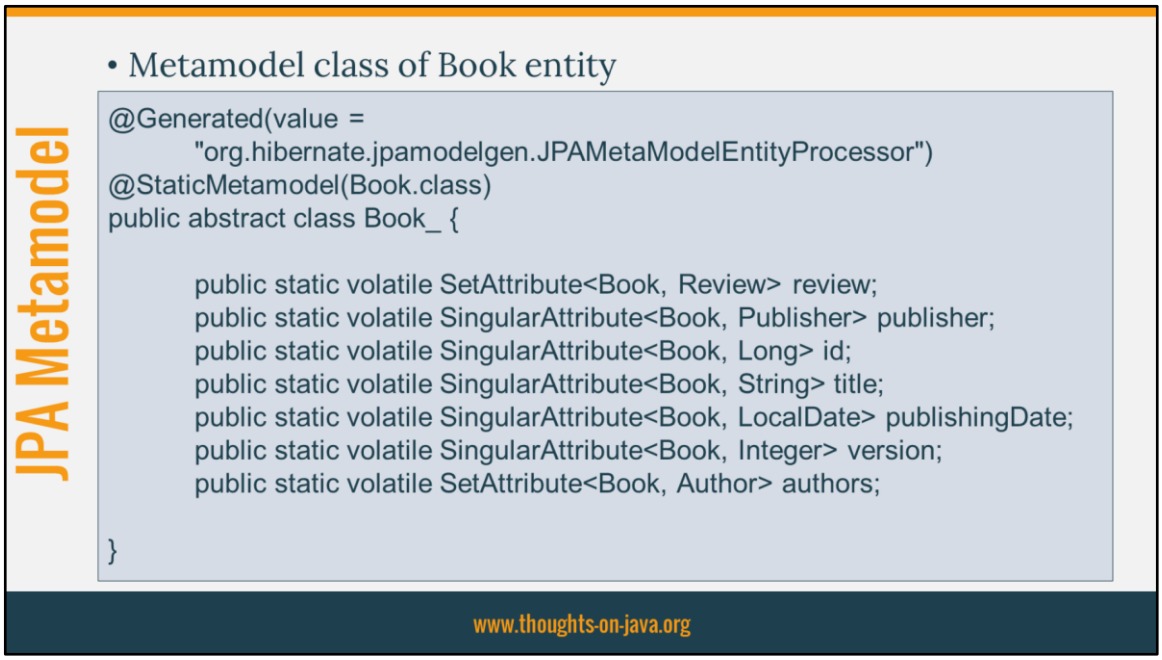


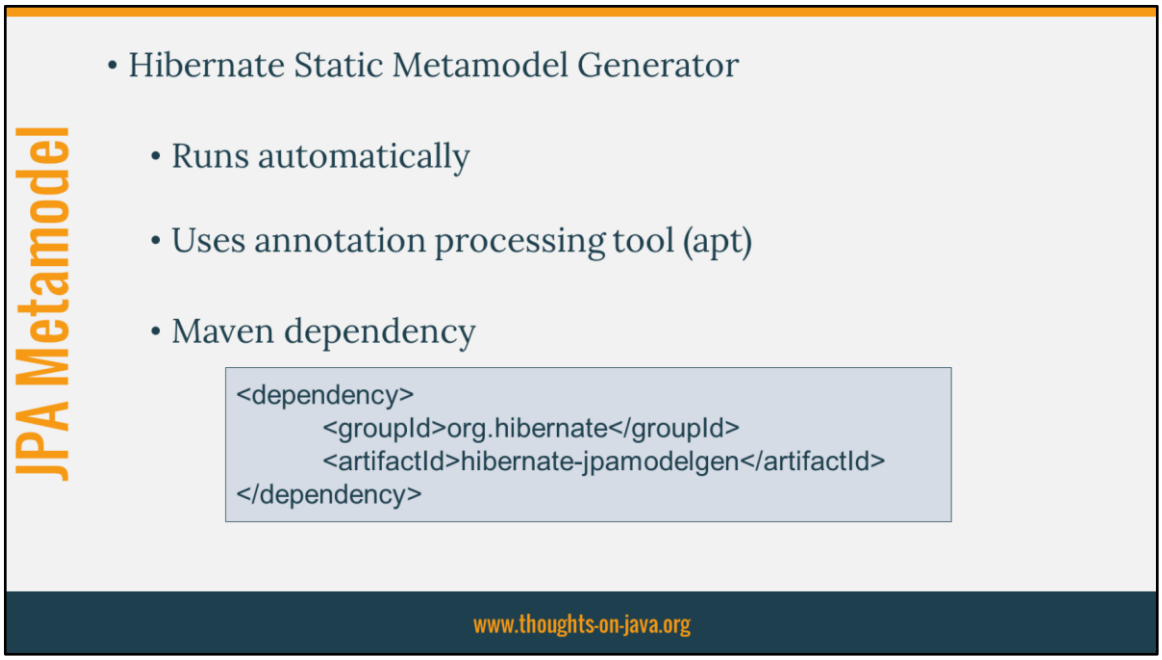
In the Criteria API video, I used to *String*s to references the entity attributes by their names. That is an easy but not a type-safe approach. If you do that in a real project, refactoring entities will become almost impossible. You would need to search for all *String*s that reference the changed attribute names and adapt them. That is an error-prone task, and you have to hope that your test cases find all the references you didn't update.
The JPA Metamodel allows you to reference all entity attributes in a type-safe way and is the recommended approach to defining Criteria queries.



The JPA Metamodel is a set of static classes that are generated at compile time. The generator creates one class for each entity with a metadata attribute for each entity attribute. You can use the metadata attributes to reference the entity attributes in a type-safe way. I will show you an example of such a class on the following slide.
The metadata classes are located in the same package as the entity classes and follow the simple naming pattern you can see on the slide. The generator just adds an '_' to the name of the entity class.



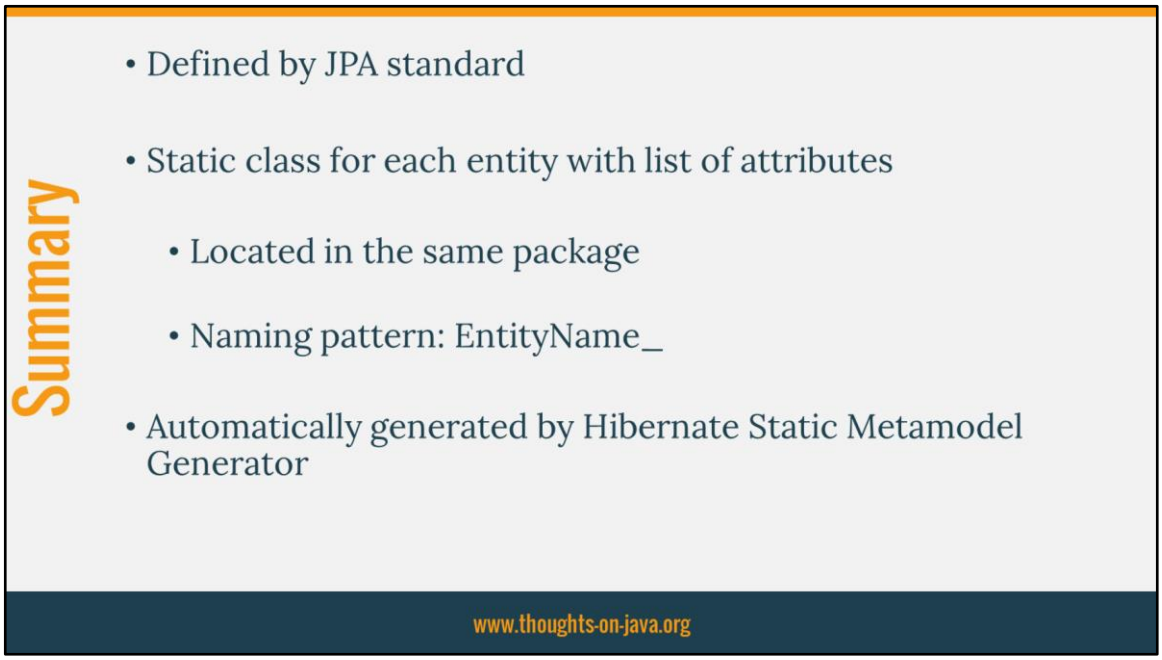
Here you can see the metadata class of the *Book* entity. It's called *Book_* and has an attribute for each entity attribute. These attributes allow you to reference all entity attributes in a type-safe way. And as you can see, each attribute is strongly typed. The first type parameter references the entity to which the attribute belongs and the second the type of the attribute.



The JPA specification recommends using the *annotation processing tool* of the Java compiler to generate the Metamodel classes. That is also what all JPA implementations do. But, unfortunately, each one provides its own solution for it. Hibernate uses its *Static Metamodel Generator* which runs automatically as a part of your build process if you add the dependency to your classpath.

Demo

www.thoughts-on-java.org



As you've seen in the example, the JPA Metamodel provides a comfortable and type-safe way to reference entity attributes. You should, therefore, prefer this approach to reference entity attributes when you define a Criteria query.
Hibernate's *Static Metamodel Generator* creates a static class for each entity which lists all of its attributes. This class is located in the same package as the entity and uses the entity name with an additional '_' as its name.

Exercises

www.thoughts-on-java.org