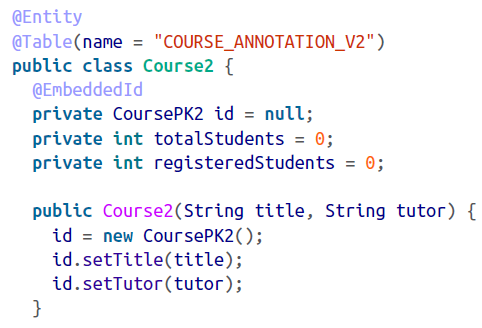
**Using Primary Key Class and @EmbeddedId**

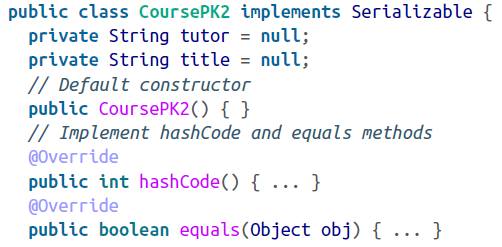
In this case, we annotate the identifier of the *Course* object with @*EmbeddedId* (instead of annotation with @*Id*   as we did in our earlier case). At the same time, we create an inner class, @CoursePK, annotating with @*Embeddedable* :



AcroRd32_2017-01-13_14-55-46.png

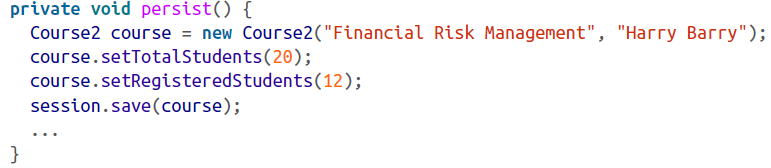
The *id* field is annotated with @*EmbeddedId*  in the preceding class. Notice that the constructor is doing the job of creating and populating the composite primary key. You can do this outside of the constructor too, if you wish.

However, we do not have to annotate @*Embeddable* on the primary key class, as we did in the first method. See the plain definition (i.e no annotation) of the primary key class *CoursePK2*  class here:



This is a simple Class that adheres to the primary composite class rules—having a default constructor, and the *hashCode* and *equals* method implementations.

Create a test Client to persist the object:



We instantiate the *Course2* object with *title* and *tutor,* which internally creates the composite key by setting these values on the *CoursePK2* object.

When you run the client, the table will be populated with the composite key and the class information, just as you would expect. The *SELECT* statement will get the same results as in the first case.