MAHARISHI UNIVERSITY of MANAGEMENT

Engaging the Managing Intelligence of Nature

Computer Science Department

CS401 Modern Programming Practices (MPP)

Professor Paul Corazza

Lecture 10:Best Programming Practices with Java 8

Living Life in Accord with Natural Law

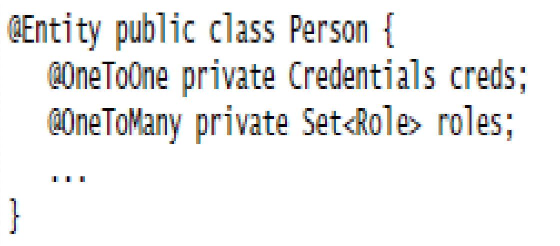
OPTIONAL: Data Access and Frameworks

Data Access and Frameworks

1. For real production code, a more systematic approach to accessing the data source is followed. These days, systems rely on framework support, which provides an interface that gives you access not only to the database, but to a robust context for working with data, including transactional support, security and access control, data visualization, and perhaps most importantly, a uniform way of reading and writing data that hides the details of creating connections and executing statements.
2. Typically, you gain access to a framework solution simply by adding one or more jar files to your project.
3. *Two Approaches.* There are two styles of framework support these days
   a. *ORM (object-relational mapping) – JPA and Hibernate use this approach*
   b. *DAOs* (data access objects) – Spring supports this approach with its JDBC templating

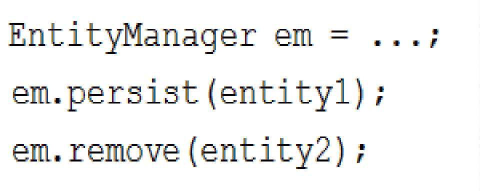
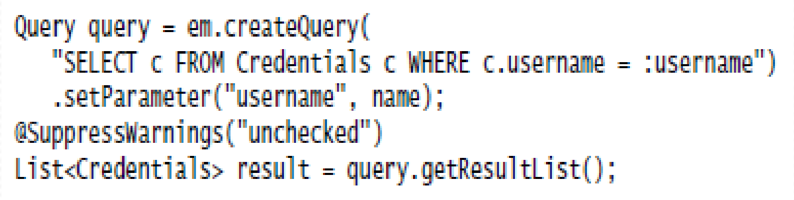
Data Access and Frameworks (cont.)

4. JPA sample.
   a. Classes that need to be persisted (like Address, Customer, etc) are called *entities*
   b. In JPA, you insert annotations in an entity class to tell the framework information about reading and writing its data.



```
@Entity public class Person {
   @OneToOne private Credentials creds;
   @OneToMany private Set<Role> roles;
   ...
}
```

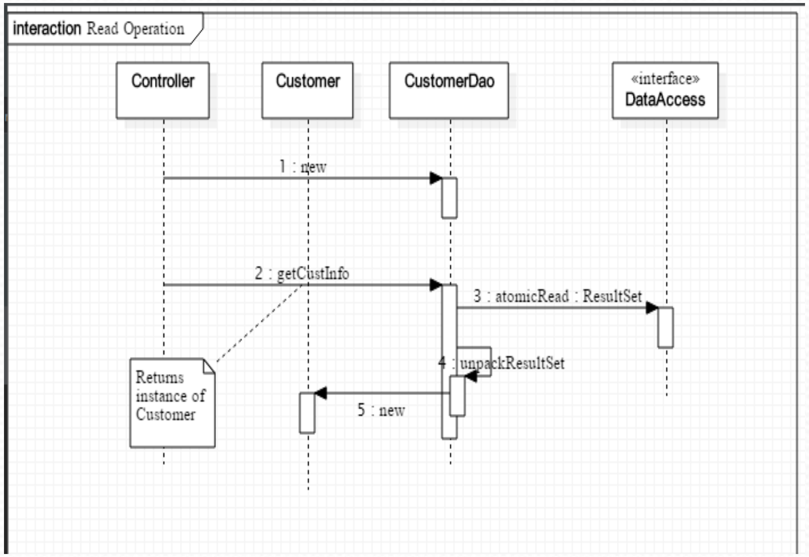
   c. An EntityManager is invoked to save entity classes to the database and also to read data from a table into one of the entity classes.

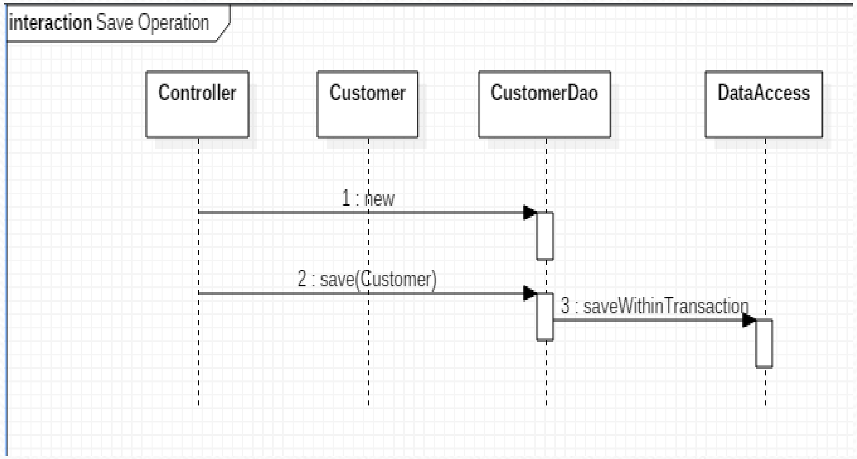

`EntityManager em = ...;`
`em.persist(entity1);`
`em.remove(entity2);`
	
	`Query query = em.createQuery(`
	`   "SELECT c FROM Credentials c WHERE c.username = :username")`
	`   .setParameter("username", name);`
	`@SuppressWarnings("unchecked")`
	`List<Credentials> result = query.getResultList();`

Persisting/Removing	**Reading**

Data Access and Frameworks (cont.)

5. DAO Sample. In the DAO approach, classes that are persistent (Address, Customer, etc) are associated with corresponding DAO classes, which know how to interact with the data access layer. For instance, a Customer class would be associated with a CustomerDao. Reads and writes of Customer are then facilitated by CustomerDao. See demo: lesson10.lecture.jdbc.framework; your must add dataaccess.jar to the project


interaction Read Operation
Controller
Customer
CustomerDao
«interface» DataAccess
1 : new
2 : getCustInfo
3 : atomicRead : ResultSet
4 : unpackResultSet
5 : new
Returns instance of Customer


interaction Save Operation
Controller
Customer
CustomerDao
DataAccess
1 : new
2 : save(Customer)
3 : saveWithinTransaction