Quiz 4: Domain-Driven Design

Due Sep 26, 2016 at 11:59pm

Points 100

Questions 5

Available Sep 21, 2016 at 8am - Sep 26, 2016 at 11:59pm 6 days

Time Limit 60 Minutes

This quiz was locked Sep 26, 2016 at 11:59pm.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	60 minutes	77.6 out of 100

Score for this quiz: **77.6** out of 100 Submitted Sep 26, 2016 at 8:32pm This attempt took 60 minutes.

Question 1 17 / 20 pts

Here is a class definition for students:

```
class Student [
  private long cwid;
  private String name;
  private String email;
  private Date dob;
  private Address address;
  private Faculty advisor;
  private List<Course> courses;
```

Annotate this with JPA annotations that specify:

- 1. The student information is in a database table called STUDENT_INFO.
- 2. The cwid field is a primary key, automatically generated by the database using the default strategy.
- 3. There is a one-to-one relationship between a student and their address, represented in the database by a field called ADDRESS_FK in the student record.
- 4. There is a one-to-many relationship between a student and their courses.
- 5. There is a many-to-one relationship between students and their faculty advisors, represented in the database by a field called ADVISOR_FK in the student record.

Make any reasonable assumptions you need to about the names of fields in other classes that map relationships, or the names of columns in database tables that represent foreign key relationships.

Your Answer:

@entity

@Table(name = Database.STUDENT_INFO)

public class Student{ @id @GeneratedValue(strategy=GenerationType.AUTO) Private long cwid private String name; private String email; private Date dob; @OneToOne (cascade=CascadeType.ALL) @JoinColumn(name = "ADDRESS_FK") public @OneToMany(fetch=FetchType.LAZY, cascade = CascadeType.ALL, mappedBy="student") private List<course> courses; @ManyToOne(fetch=FetchType.LAZY) @JoinColumn(name="ADVISOR_FK") private Faculty advisor; }

@JoinColumn(name="ADDRESS_FK") private Address address;

	Question 2	6.6 / 20 pts
	For each of the relationships between entities in the Student entity that "owns" the relationship. Identify the owning iden that entity:	
	1. Student-Address public class student	
	2. Student-Advisor	
	3. Student-Course	
	Answer 1:	
You Answered	public class student	
orrect Answer	Student	
	Answer 2:	
You Answered	(You left this blank)	
orrect Answer	Student	
	Answer 3:	
You Answered	(You left this blank)	
orrect Answer	Course	

Question 3 20 / 20 pts

What is the purpose of each of the following two design patterns in ORM:

- 1. Persistence Data Object (PDO)
- 2. Data Access Object (DAO)

Your Answer:

You Answered

Correct Answer

Answer 2:

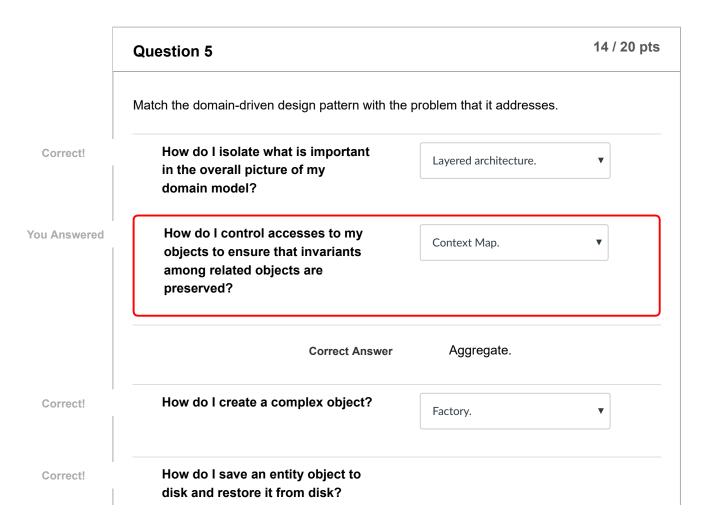
1) Persistent Data Object:

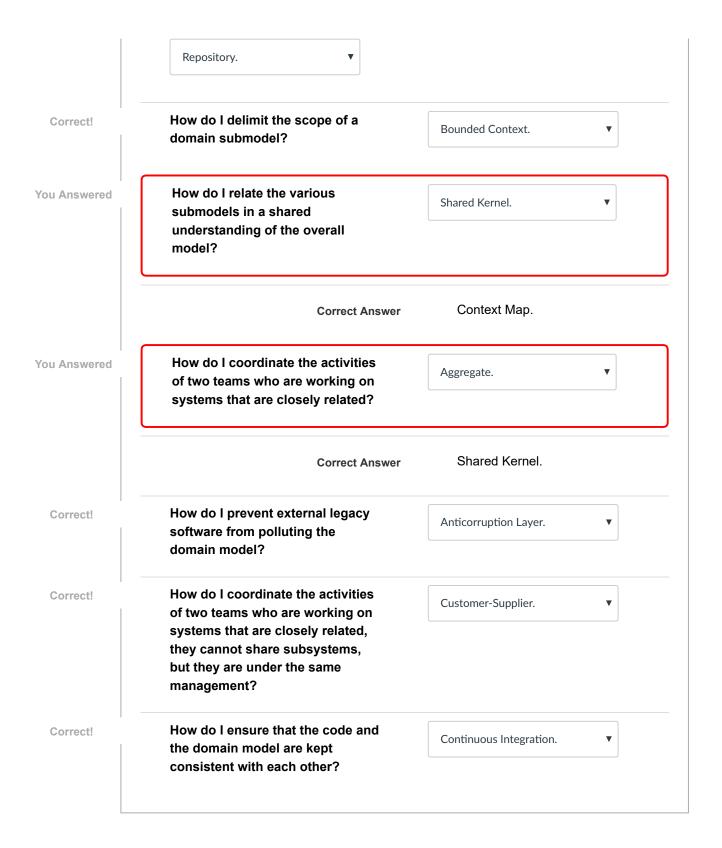
encapsulating domain logic in domain objects is a fundamental precept of the DDD methodology. The repository pattern is used to save domain objects, with their state, in databases. Persistent framework mostly uses ORM to interface to user code and covers the problamatic to make the storahe of objects as secure and reliable as possible.

2) Data Access Object: this is an object that proviudes an abstract interface to some type to some type of database or other persistence mechanism. By mapping application calls to the persistence layer, DAO provide some specific data operations without exposing details of the database.

Question 4	20 / 20 pts			
Give the names of the JPA EntityManager methods for each of the following tasks: 1. add an entity to the managed objects entityManager.persist()				
	Manager.remove(
3. flush updates on an entity to the database	entityManager.flush()			
4. detach an entity from the managed objects	void detach(Object enti			
5. re-attach an entity to the managed objects	entityManager.merge()			
Answer 1: entityManager.persist()				
persist				

You Answered	entityManager.remove()
Correct Answer	remove
	Answer 3:
You Answered	entityManager.flush()
Correct Answer	flush
	Answer 4:
You Answered	void detach(Object entity)
Correct Answer	detach
	Answer 5:
You Answered	entityManager.merge()
	merge





Quiz Score: 77.6 out of 100