**BẢNG ĐÁNH GIÁ ĐIỂM THUYẾT TRÌNH**

**MÔN LẬP TRÌNH JAVA NÂNG CAO**

**LỚP CTK44-PM**

|  |  |
| --- | --- |
| **Thông tin đề tài thuyết trình** | **Thông tin sinh viên thực hiện đánh giá** |
| Tên đề tài: Tìm hiểu về Lập trình JPA (Java Persistence API) | MSSV của sinh viên đánh giá: 2012353 |
| Điểm tổng kết: Ngày trình bày: 11/04/2023 | Họ tên sinh viên & ký xác nhận: Đoàn Cao Nhật Hạ |
| ***Thông tin sinh viên thuyết trình*** |
| MSSV: 2011439  Họ tên: Huỳnh Tấn Thanh |

1. **ĐÁNH GIÁ THUYẾT TRÌNH**

**Thang điểm: Kém 0-5; Trung bình: 5-7; Khá 7-8; Tốt 8-9; Xuất sắc: 9-10**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bảng chấm điểm nhóm thuyết trình** | | | | | | |
| Các chủ đề chính có được xác định? (A) | Slide thuyết trình có thiết kế tốt (B) | Giao tiếp với người nghe bằng ánh mắt (C) | Giao tiếp với người nghe bằng giọng nói (D) | Khả năng trình bày nội dung (E) | Thảo luận & Hỏi đáp (F) | Điểm trung bình = (A+B+C+D+E+F)/6 |
| **8** | **8** | **8** | **8** | **8** | **8** | **8** |

1. **BÀI TẬP NẮM KIẾN THỨC**

Dựa vào nội dung vừa trình bày, trả lời các câu hỏi sau:

1. Trình bày tóm tắt nội dung anh/ chị đã nắm được về chủ đề vừa trình bày?

JPA là viết tắt của **Java Persistence API**:

* Là một đặc tả Java cho việc ánh xạ giữa các đối tượng Java với cơ sở dữ liệu quan hệ sử dụng công nghệ phổ biến là **ORM** (Object Relational Mapping).
* Cung cấp đầy đủ các công cụ cho phép chúng ta có thể thao tác với cơ sở dữ liệu một cách đơn giản và nhanh chóng. JPA có thể dùng để **persist** một đối tượng Java (**POJO** – Plain Old Java Object) vào trong cơ sở dữ liệu hoặc lấy dữ liệu từ cơ sở dữ liệu và ánh xạ (**mapping**) ra các đối tượng Java một cách đơn giản.
* Hoạt động như một **cầu nối** giữa các table/ các mối quan hệ giữa các table trong database và các class/ mối quan hệ giữa các object.

ORM là viết tắt của Object Relational Mapping:

* Là một công nghệ/ khái niệm/ quá trình chuyển đổi dữ liệu từ ngôn ngữ hướng đối tượng sang Database quan hệ và ngược lại
* Có khả năng xử lý các thao tác của nhiều loại cơ sở dữ liệu khác nhau một cách dễ dàng mà không quan tâm đến loại database sử dụng (SQL Server, MySQL, PostgreSQL, …) hay loại thao tác sử dụng (INSERT, UPDATE, DELETE, SELECT, …).

Một số ORM framework có thể sử dụng cho JPA như: Hibernate, iBatis, Eclipse Link, OpenJPA, TopLink,….

* Hibernate: là một ORM framework cho phép lập trình viên thao tác với database hoàn thành tự nhiên thông qua các đối tượng, lập trình viên không cần quan tâm đến loại database sử dụng SQL,….

Lợi ích:

* Đơn giản hóa công nghệ cho tầng persistence (tầng dữ liệu).
* Không phụ thuộc vào các framework ORM.
* Có nhiều nhà cung cấp hỗ trợ cài đặt JPA.
* Dữ liệu có thể được lưu trữ thông qua việc ORM.

Kiến trúc JPA:

* Sử dụng metadata để ánh xạ các đối tượng persistence với các bảng trong cơ sở dữ liệu. JPA hỗ trợ SQL như là một ngôn ngữ truy vấn để dễ dàng xử lý các truy vấn cơ sở dữ liệu. Ngôn ngữ truy vấn JPA có thể dùng thực thi cả truy vấn tĩnh và truy vấn động.
* Gồm ba thành phần chính là: **Entity, EntityManager** và **EntityManager**Factory. Ngoài ra còn có EntityTransaction, Persistence, Query.
* **Entity**: là các đối tượng thể hiện tương ứng 1 table trong cơ sở dữ liệu. Entity thường là các class POJO đơn giản, chỉ gồm các phương thức getter, setter.
* Một số đặc điểm của một Entity:

+ Entity có thể tương tác với cơ sở dữ liệu quan hệ.

+ Entity được xác định thông qua một định danh (id), tương đương với khóa chính trong table của cơ sở dữ liệu quan hệ.

+ Entity hỗ trợ transaction.

+ Entity hỗ trợ kế thừa giống như những class Java khác.

* **EntityManager:** là một interface cung cấp các API cho việc tương tác với các Entity.
* Một số chức năng cơ bản của EntityManager như:

+ Persist: phương thức này dùng để lưu một thực thể mới tạo vào cơ sở dữ liệu.

+ Merge: dùng để cập nhật trạng thái của entity vào cơ sở dữ liệu.

+ Remove: xóa một instance của entity.

* **EntityManagerFactory:**  được dùng để tạo ra một instance của EntityManager.
* **EntityTransaction**:
* Một tập hợp các thao tác trong đó tất cả các thao tác phải được thực hiện thành công hoặc tất cả thất bại.
* Một database transaction bao gồm một tập hợp các câu lệnh SQL được committed hoặc rolled back trong một unit.
* EntityTransaction có quan hệ 1-1 với EntityManager. Bất kỳ thao tác nào được bắt đầu thông qua đối tượng EntityManager đều được đặt trong một Transaction. Đối tượng EntityManager giúp tạo EntityTransaction.
* **Persistence**:
* Định nghĩa một tập hợp các Entity class được quản lý bởi 1 instacne của EntityManager trong ứng dụng.
* Persistence (javax.persistence.Persistence) class bao gồm các phương thức static để lấy instance của EntityManagerFactory.
* **Query**: là một interface, được mỗi nhà cung cấp JPA implement để có được các đối tượng quan hệ đáp ứng các tiêu chí (criteria) truy vấn.

Một số tính năng của JPA:

* JPA hỗ trợ plugable, tức là có thể sử dụng nhiều 3rd khác nhau như Hibernate hay MyBatis.
* Hỗ trợ cấu hình thông qua annotation và xml.
* Giảm bớt số lớp yêu cầu cho việc phát triển persistence.
* Không cần phải viết các mô tả triển khai trong XML. Các Annotation dựa trên metadata đã hỗ trợ trong các ứng dụng JPA.
* Đã chuẩn hóa ORM và dễ dàng phát triển hơn.
* JPA hỗ trợ truy vấn động và tĩnh.
* Nhiều IDE hỗ trợ phát triển ứng dụng JPA và có thể tự động sinh code ánh xạ từ cơ sở dữ liệu thành các entity và ngược lại.

1. Anh/ chị hãy đưa ra các nhận xét và góp ý cho sinh viên vừa trình bày?

Cần giải thích rõ hơn về phần demo kết hợp sản phẩm