1. **To meet the system's maintenance requirements, every customer request must be recorded in a new table. Please write JPA objects (including Service, Dao, Entity).** **Table Name : execution\_record**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Column Type** | **Not Null** | **Description** |
| **serno** | **Int8** | **Y** | **PK** |
| **serviceName** | **Varchar** | **Y** |  |
| **msgid** | **Varchar** | **Y** |  |
| **requestPayload** | **Varchar** | **Y** |  |
| **request\_timestamp** | **Timestamp** | **Y** |  |

Ans:

|  |
| --- |
| ExecutionRecordEntity.java  {  @Id  @GeneratedValue(strategy = GenerationType.IDENTITY)  Private int serno;  Private String serviceName;  Private String msgid;  Private String requestPayload;  Private Timestamp requestTimestamp;  List of setter and getter methods;  }  ExecutionRecordDAO.java  {  @PersistenceContext  private EntityManager entityManager;  public void save(ExecutionRecordEntity entity){  entityManager.persist(entity);  }  }  ExecutionRecordService.java  {  @Autowire  Private ExecutionRecordDAO recordDao;  @Transaction  Public void recordCustomerRequest(String serviceName, String requestPayload, string msgid) {  ExecutionRecordEntity record = new ExecutionRecordEntity();  Record.setServiceName(serviceName);  Record.setMsgid(msgid);  Record.setRequestPayload(requestPayload);  Record.setRequestTimestamp(new Timestamp());  recordDao.save(record);  }  } |

1. **A new exception for APIs is being requested. In order to reduce costs and unexpected risk, please write down an AOP code to handle this exception, and return HTTP Error code 409 Conflict. APIs package location : com.testexam.controller ConflictException.java**

|  |
| --- |
| **public class ConflictException extends RuntimeException {**  **private static final long serialVersionUID = 1L;**  **private String returnCode;**  **private String returnDesc;**  **public ConflictException(String returnCode) {**  **this.returnDesc = getMessageDesc(returnCode);**  **this.returnCode = returnCode;**  **}**  **public String getReturnCode() {**  **return this.returnCode;**  **}**  **private String getMessageDesc(String corpusCode) {**  **String desc = "";**  **switch (corpusCode) {**  **case "A":**  **this.returnDesc = "MESSAGE-A";**  **break;**  **case "B":**  **this.returnDesc = "MESSAGE-B";**  **break;**  **default:**  **this.returnDesc = "MESSAGE-D";**  **}**  **return desc;**  **}**  **public String getReturnDesc() {**  **return returnDesc;**  **}**  **public String getReturnCode(String returnCode) {**  **return returnCode;**  **}**  **}** |

**Ans**:

|  |
| --- |
| @ControllerAdvice  @Slf4j  @EnableWebMvc  public class ExceptionHandler {  @ExceptionHandler(ConflictException.class)  public ResponseEntity<String> handleApiException(  ConflictException ex, ServletWebRequest request) {  log.error(ex.getMessageDesc());  return ResponseEntity.status(409)  .body(ex.getMessageDesc());  }  } |

1. **Restructure the code below to decrease the interdependence between the classes. Consider applying appropriate design patterns, such as dependency injection or the factory pattern, to achieve a more loosely coupled and maintainable design by using Spring framework features.**

|  |
| --- |
| **class Zoo {**  **Animal tiger;**  **Animal zebra;**  **public Zoo() {**  **tiger = new Giraffe();**  **zebra = new Panda();**  **tiger.eat();**  **zebra.eat();**  **}**  **}**  **interface Animal {**  **public void eat();**  **}**  **class Giraffe implements Animal {**  **public void eat() {**  **System.out.println("Giraffe eat grass.");**  **}**  **}**  **class Panda implements Animal {**  **public void eat() {**  **System.out.println("Panda eat bamboo.");**  **}**  **}** |

**Ans**:

|  |
| --- |
| **class Zoo {**  **Animal tiger;**  **Animal zebra;**  **public Zoo(Animal tiger, Animal zebra) {**  **this.tiger = tiget;**  **this.zebra = zebra;**  **}**  **Public eat() {**  **this.tiger.eat();**  **this.zebra.eat();**  **}**  **}**  **interface Animal {**  **public void eat();**  **}**  **class Giraffe implements Animal {**  **public void eat() {**  **System.out.println("Giraffe eat grass.");**  **}**  **}**  **class Panda implements Animal {**  **public void eat() {**  **System.out.println("Panda eat bamboo.");**  **}**  **}** |

1. **To guarantee thread safety, modify the code below and use the appropriate synchronization mechanisms or concurrent utilities provided by the Java standard library or the Spring framework. MyBeam.java**

|  |
| --- |
| **import java.util.Map;**  **import org.springframework.core.env.Environment;**  **import com.testexam.dao.QueryDAO;**  **import com.testexam.dao.RequestObject;**  **public class Calculation {**  **private static Calculation uniqueInstance;**  **private Environment env;**  **private QueryDAO dao;**  **public Calculation(QueryDAO dao, Environment env) {**  **this.env = env;**  **this.dao = dao;**  **}**  **}** |

Ans:

|  |
| --- |
| **import java.util.Map;**  **import org.springframework.core.env.Environment;**  **import com.testexam.dao.QueryDAO;**  **import com.testexam.dao.RequestObject;**  **public class Calculation {**  **private static Calculation uniqueInstance;**  **private Environment env;**  **private QueryDAO dao;**  **private Calculation(QueryDAO dao, Environment env) {**  **this.env = env;**  **this.dao = dao;**  **}**  **Public static synchronized getInstance(QueryDAO dao, Environment env) {**  **If (uniqueInstance == null) {**  **uniqueInstance = new Calculation(dao, env);**  **}**  **Return uniqueInstance;**  **}**  **}** |