| 1 | 다음 설명 중 틀린 것을 모두 고르면?  (1)자바의 데이터베이스 연동기술에는 jdbc,ibatis/mybatis , hibernate, jpa 등이 있다.  (2)Hibernate는 SQL문과 자바 객체와의 매핑을 지원하는 SQL Mapping 프레임웍이다.  (3)MyBatis는 Table과 자바 객체와의 매핑을 지원하는 ORM 프레임웍이다.  (4)JPA란 javaEE 자체에서 지원하는 ORM API 다. |
| --- | --- |
| 2 | 다음 설명 중 틀린 것은?  (1)자바 언어를 이용하여 데이터베이스를 연동하기 위한 기술을 JDBC라 한다  (2)현존하는 거의 모든 DBMS는 JDBC로 제어할 수 있어야 하므로 Sun에서는 정기적으로 전세계의 DBMS  벤더사를 조사하여 JDBC 코드를 작성하고 있다.  (3)Sun에서는 자바 언어로 DBMS를 제어할 수 있는 API의 기준만 세워 놓은 것이고, 실제 해당 DBMS에 최적  화된 코드를 작성할 의무는 DBMS를 팔아먹는 회사에게 있다  (4)(3)번과 같이 해당 DBMS 회사 제품에 맞게 구현된 코드 구현체를 드라이버라 하며 DBMS 벤더가 관리 및  배포한다 |
| 3 | 다음 설명 중 틀린 것을 모두 고르면?  (1)자바 언어를 이용하여 데이터베이스를 연동하려면, 연동하려는 DBMS에 대한 드라이버를 개발환경에  빌드해 놓아야 한다.  (2)자바 언어가 데이터베이스에서 커넥션을 취득하려면 결국 JDBC 드라이버가 없으면 불가능하다.  (3)자바의 JDBC를 이용하여, 커넥션을 취득하는 방법은 요청이 있을 때마다 취득하는 방법이 있고, 요청이  없어도 미리 취득해 놓는 방법이 있는데, 특히 후자의 경우를 커넥션 풀링이라 한다.  (4)커넥션 풀링의 원리를 분석하면 개발자가 직접 작성할 수도 있지만 일반적으로 검증된 커넥션 풀을 선호  하므로 독립실행형 어플리케이션일 경우 아파치의 DBCP 등을 많이 사용한다. |
| 4 | 다음 설명 중 틀린 것을 모두 고르면?  (1)JNDI란 커넥션풀을 의미한다.  (2)JNDI란 자바 코드 내에 자원 정보를 두지 않고, XML과 같은 외부 설정 파일로 분리시켜 정보를 관리하기  위한 javaSE의 자원 관리 API 일뿐이다.  (3)JNDI는 특히 데이터베이스 연결 정보와 관련되어서 자주 사용되는 것일 뿐 JNDI 자체가 커넥션 풀은  아니다  (4)WAS나 웹컨테이너(톰켓,레진 등) 서버는 JNDI를 이용한 커넥션 관리 기능을 지원하는데, 특히 이들 서버  에서 지원하는 JNDI는 커넥션 풀링 기능이 기본으로 포함되어 있을 뿐, JNDI가 커넥션 풀 자체는 아니다  (5)JNDI를 사용하면 JDBC를 사용하지 않고도 데이터베이스에 접속할 수 있다. |
| 5 | 다음 중 틀린 것은?  (1)스프링은 Enterprise 분야에서 필요시 되는 트랜잭션 처리, 보안 등을 지원하고 있다.  (2)트랜잭션을 지원하려면 데이터베이스 커넥션 제어는 스프링이 주도할 수 있어야 한다.  (3)스프링이 지원하는 데이터베이스 기술을 스프링 JDBC라 한다.  (4)Mybatis나 Hibernate 를 이용하면 JDBC를 사용할 필요 없다.  (5)Mybatis와 같은 SQL Mapper나 Hibernate, JPA와 같은 ORM 기술은 JDBC를 배제하려는 기술이 아니라  JDBC로 직접 구현할 때의 상투적 코드의 비효율성을 보다 편리하게 대신해 줄 뿐이다. |
| 6 | 다음 설명 중 틀린 것은?  (1)JDBC API를 직접 이용하여 개발자가 insert 쿼리를 시도하려면 이미 DTO 객체 안에 들어있는 멤버변수  값을 일일이 끄집어 내어 SQL문과 짝을 맞추는 행위를 반복하게 된다.  (2)실제 쿼리문 코드보다, 커넥션을 얻고 닫는 과정에 필요한 코드가 너무 거창하며, 반복적으로 수행된다.  (3)JDBC API에서 지원되는 객체들을 직접 사용하게 되면, 결국 개발자는 쿼리문과 자바 객체와의 매핑을  일일이 반복 수행해야 한다  (4)위 문제들을 해결하기 위해 개발된 퍼시스턴스 영역의 프레임워크들이 많이 개발되어 있다.  (5)자바 객체와 테이블간의 매핑을 자동으로 처리해주는 퍼시스턴스 프레임웍에는 Mybatis가 있다. |
| 7 | 다음 설명 중 틀린 것은?  (1)스프링도 데이터베이스 연동을 지원하기 위해서는 JDBC 사용은 피해갈 수 없다. 즉 JDBC 기반 위에서  개발자를 위한 기능들을 지원하고 있다고 보아야 한다.  (2)스프링은 개발자가 직접 JDBC API 관련 객체들(Connection, PreparedStatement, Result 등)을 다루기를  장려한다.  (3)스프링은 개발자가 로직에 집중할 수 있도록 데이터베이스 연동 기술을 지원하고 있고, 특히 JDBC를  직접 제어하지 않도록 다양한 템플릿을 지원하고 있다.  (4)개발자가 선호하는 기술이 JDBC API라면 스프링의 JdbcTemplate 을 사용할 수 있다.  (5)개발자가 선호하는 기술이 Mybatis 라면 스프링의 SqlSessionTemplate을 사용할 수 있다.  (6)개발자가 선호하는 기술이 Hibernate 라면 스프링의 HibernateTemplate 을 사용할 수 있다. |
| 8 | 다음 설명 중 틀린 것은?  (1)개발자가 작성하는 DAO는 데이터베이스 연동 기술의 종류에 최적화되고 종속적일 수록 유지보수성이  높다.  (2)개발자가 선호하는 기술이 무엇이던 DAO 코드는 크게 영향을 받지 않아야 한다.  (3)(2)번의 이유로 인해 스프링은 각 기술마다 대응되는 템플릿 클래스를 지원한다.  (4)스프링의 템플릿을 이용하지 않으면 트랜잭션을 지원받을 수 없다.  (5)스프링으로 인해 앞으로 JDBC 기술은 사라질 것이다.  (6)DataSourceUtils 객체를 이용하여 Connection 객체를 직접 취득할 경우는 스프링의 트랜젹션을 지원받을  수있다. |
| 9 | 빈칸에 알맞는 클래스명을 채우세요   | **DB연동 기술** | **지원 Template 클래스** | **지원 DaoSupport 클래스** | | --- | --- | --- | | Jdbc |  |  | | Mybatis |  |  | | Hibernate |  |  | | JPA |  |  | |
| 10 | 스프링의 트랜잭션과 관련된 설명 중 틀린 것은?  (1)스프링을 이용하면 연동 기술마다 트랜잭션을 처리하는 방식과 코드를 달리 할 수 있다.  (2)스프링을 이용하면 개발자가 Hibernate를 이용하던, Mybatis를 이용하던, Jdbc를 이용하던 트랜잭션을  처리하는 코드에는 변함이 없다.  (3)(2)와 같은 특성은 스프링이 TransactionManager 를 지원하기 때문이다.  (4)스프링은 각각의 연동 기술에 대응되는 TransactionManager를 지원한다.  (5)트랜잭션이 처리될 때 TransactionManager 는 각각의 연동 기술에 트랜잭션을 전파하여 트랜잭션을  처리한다. |
| 11 | 데이터 연동기술에 따라 알맞는 PlatformTransactionManager 구현 클래스를 채우세요.   | **연동 기술** | **구현 클래스** | | --- | --- | | JDBC |  | | iBatis / MyBatis |  | | Hibernate |  | | JPA |  | |
| 12 | 다음 설명 중 틀린 것을 고르면?  (1)스프링은 데이터베이스 연동 뿐 아니라 트랜잭션도 지원하고 있는데, 특히 트랜잭션 관련되어서는 선언적  트랜잭션만을 지원한다.  (2)스프링은 여러 유형의 데이터 연동 기술을 지원하고 있으며 DB연동 기술에 따라 각각 틀린 코드로 트랜  잭션을 관리한다. 즉 데이터 연동 플랫폼에 따라 트랜잭션 제어 코드가 서로 틀리다.  (3)스프링은 PlatformTranactionManager 인터페이스를 지원하므로, 데이터베이스 연동 기술이 틀리더라도  동일한 코드로 트랜잭션을 관리할 수 있도록 지원한다.  (4)일반적으로 트랜잭션을 처리하는 계층은 DAO이다. |
| 13 | 스프링의 트랜잭션과 관련된 설명 중 틀린 것은?  (1)코드를 이용한 트랜잭션 처리와 XML을 이용한 선언적 트랜잭션 모두를 지원한다.  (2)코드를 이용한 트랜잭션 처리시 사용되는 클래스는 TransactionTemplate이다.  (3)<tx:advice> 태그를 이용한 트랜잭션 처리는 선언적 방법이다.  (4)@Transactional 어노테이션을 이용한 트랜잭션 처리는 코드적 트랜잭션 방법이다. |