
[문제 1] Spring Framework 의 가장 기본이 되는 개념이다. 많은 Spring 모듈들이 이 방법을 사용해서 개발되고 있다. 클래스간 강한 결합을 제거해주고, 유연한 앱을 제작할 수 있도록 해준다. 이 방법은 무엇인가?

- ① Dependency Injection
- ② Exception Handling
- ③ AOP
- ④ Service Abstraction

정답: ①

해설: 스프링의 핵심 원리 중 하나로, 객체 간의 결합도를 낮추기 위해 의존성을 외부에서 주입하는 개념이다.

[문제 2] 다음 중 스프링 프레임워크에 대한 설명 중 옳지 않은 것은?

- ① 자바 엔터프라이즈 개발을 편하게 해주기 위해서 만들어졌다
- ② EJB 와 같은 Heavy-weight 프레임워크가 아니라 POJO 를 사용해서 개발이 가능한 경량급 프레임워크이다.
- ③ 고급 웹 컨테이너를 사용하면 엔터프라이즈 개발의 고급기술을 대부분 사용할 수 있다.
- ④ 스프링은 오픈 소스의 장점을 충분히 취하면서 동시에 오픈 소스 제품의 단점과 한계를 잘 극복하고 있다.

정답: ③

해설: 스프링은 고급 웹 컨테이너 없이도 고급 엔터프라이즈 기술을 대부분 지원한다.

[문제 3] Point Cut, Advisor 등으로 Weaving 을 통해 '어느 때에 어느 공통된 기능을 실행한다'라는 방식의 프로그래밍은 무엇인가?

- ① Dependency Injection
- ② Aspect Oriented Programming
- ③ Spring MVC Framework
- ④ Inversion of Control

정답: ②

해설: AOP 는 공통 관심사를 모듈화하여, 비즈니스 로직과 분리하여 유지보수를 쉽게 한다.

[문제 4] 요청 구조상 Dispatcher Servlet 앞에 위치하는 것으로 요청에 대한 전반적인 정보를 가지고 있다. CharSet 을 변경하거나 세션 인증에 주로 쓰이는 것은 무엇인가?

- ① Interceptor
- ② Bean Context
- ③ Filter
- ④ Logback

정답: ③

해설: Filter 는 DispatcherServlet 앞단에서 요청 전반을 제어하기 위해 주로 사용된다.

[문제 5] 이것은 기능 수정 및 리팩토링 후에 발생할 수 있는 사이드 이펙트를 최소화 시킬 수 있다. 정상, 실패, 예외상황에 대해 검증할 수 있는 테스트 프레임워크는 무엇인가?

- ① junit
- ② AOP
- ③ Maven
- ④ Logback

정답: ①

해설: junit 은 자바의 단위테스트 프레임워크로, 코드의 정상 및 실패 케이스를 검증한다.

[문제 6] Spring MVC Application 에서 발생하는 Exception 을 통합 처리할 수 있는 Annotation 은 무엇인지 고르시오.

- ① @ExceptionHandler
- ② @TryCatch
- ③ @ControllerAdvice
- ④ @RequestMapping

정답: ③

해설: @ControllerAdvice 는 컨트롤러 전역에서 발생하는 예외를 처리하는 데 사용되는 어노테이션이다.

[문제 7] 모든 페이지마다 동일한 기능의 제어를 목적으로 만들어진 기능이며, 컨트롤러 실행 전, 컨트롤러 실행 후, 뷰 실행 후 등 3 가지 부분을 컨트롤 할 수 있다. 주로 요청의 처리시간, 인증 체크, 인가 체크에 사용되는 이것은 무엇인지 고르시오.

- ① Filter
- ② Interceptor
- ③ AOP
- ④ Dependency Injection

정답: ②

해설: Interceptor 는 컨트롤러의 실행 전·후, 뷰 렌더링 후 등 3 단계에서 공통 작업을 처리할 수 있다.

[문제 8] 다음 중 MyBatis 의 동적 쿼리(Dynamic Query)에서 사용할 수 있는 태그가 아닌 것은?

- ① if
- ② choose, when
- ③ foreach
- ④ While

정답: ④

해설: MyBatis 의 동적 쿼리 태그에는 if, choose, foreach 등이 있고 while 은 없다.

[문제 9] 다음은 MyBatis 동적 쿼리(Dynamic SQL)의 일부분이다. 빈칸에 알맞은 단어를 선택하시오.

PET_NAME= #{name}

- ① test="name !=null"
- ② "name !=null"
- ③ test=name !=null
- ④ check=(name !=null)

정답: ①

해설: MyBatis 의 if 조건문에서는 test 속성을 사용한다.

[문제 10] 다음 중 Convention over Configuration 을 설명한 것 중 잘못된 것은?

- ① 설정보다는 관행에 따라 미리 정해진 설정으로 개발을 수행하는 것을 선호한다는 의미이다.
- ② 널리 알려진 best practice 에 따라 디렉토리 구조나 소스코드 일부를 생성해준다.
- ③ 메이븐이 이 방식에 따라 만들어 졌다.
- ④ 라이브러리에 대한 의존성 관리를 별도로 해준다.

정답: ④

해설: 라이브러리 의존성 관리는 Maven 의 기능이지만, Convention over Configuration 의 정의와는 다르다.

[문제 11] 다음 중 IoC 의 개념을 설명할 것 중 옳지 않은 것은?

- ① 객체의 생성, 생명주기의 관리까지 모든 객체에 대한 제어권이 바뀌었다는 것을 의미한다.
- ② IoC 컨테이너는 객체의 생성을 책임지고 의존성을 관리한다.
- ③ 스프링 IoC 를 사용하면 개발자가 직접 객체를 생성할 수 없다.
- ④ POJO 의 생성, 초기화, 서비스, 소멸에 대한 권한을 가진다.

정답: ③

해설: 스프링 IoC 를 사용해도 개발자가 직접 객체를 생성할 수는 있다.

[문제 12] 다음 중 각 레이어에 대한 설명 중 옳지 않은 것은?

- ① 비즈니스 레이어 : 기능을 수행, 트랜잭션 수행
- ② 데이터 영역 : 데이터의 저장과 조회를 담당
- ③ 프레젠테이션 레이어 : 화면의 효과를 담당
- ④ 데이터 영역 : 주로 데이터베이스와 연동하여 작업

정답: ③

해설: 프레젠테이션 레이어는 사용자와 상호작용하며, 화면의 효과가 아니라 UI 를 담당한다.

[문제 13] 요청 파라미터의 값이 없어도 에러를 보내지 않고 처리가 가능한 코드는 무엇인가?

- ① `@RequestParam(value="id", required=false, defaultValue="")`
- ② `@RequestParam(value="id", required=true, defaultValue="")`
- ③ `@RequestParam(value="id", required="false", defaultValue="")`
- ④ `@RequestParam(value="id", required="true", defaultValue="")`

정답: ①

해설: `required=false` 로 설정하면 파라미터가 없어도 에러를 내지 않는다.

[문제 14] AOP 대한 설명 중 잘못된 것은?

- ① Aspect Oriented Programming 의 약어이다.
- ② Advice 와 point cut 을 합친 용어를 Aspect 라고 한다.
- ③ 스프링에서는 독자적으로 개발한 스프링 Aspect 기술을 사용하여 구현한다.
- ④ 핵심기능과 부가기능을 분리하기 위해 사용하는 기술이다.

정답: ③

해설: 스프링은 AspectJ 의 일부를 스프링 AOP 로 지원하며, 독자적으로 Aspect 를 개발한 것은 아니다.

[문제 15] 포인트컷 표현식 문법에 대한 설명 중 틀린 것은?

- ① AspectJ 포인트컷 표현식은 포인트컷 지시자를 이용하여 작성한다.
- ② 포인트컷 지시자 중에서 가장 대표적으로 사용되는 것은 `execution()`이다.
- ③ 표현식의 문법구조 중 메서드 개수에 상관없이 정의는 불가능하다.
- ④ 표현식의 문법구조 중 접근제한자 별로 정의가 가능하다.

정답: ③

해설: 와일드카드를 사용하면 메서드 개수에 상관없이 정의 가능하다.

[문제 16] 스프링 AOP 구현방식에 대한 설명 중 옳은 것은?

- ① `@Aspect` 어노테이션을 사용하여 어드바이스와 포인트컷이 한번에 정의하여 개발할 수 있다.
- ② `@Aspect` 어노테이션을 사용하려면 `proxy-targetClass` 를 설정해 주어야 한다.
- ③ XML 기반의 AOP 구현은 지원하지 않는다.
- ④ Advice 모두 3 가지 있으면 Around 어드바이스가 가장 많이 사용된다.

정답: ①

해설: `@Aspect` 를 사용해 Advice 와 Pointcut 을 한 클래스에 선언할 수 있다.

[문제 17] 어드바이스에 대한 설명으로 옳지 않은 것은?

- ① @Before : 타깃 객체의 메서드가 호출된 직후 호출되는 어드바이스
- ② @After : 타깃 객체의 메서드가 정상 종료되었을 때와 예외가 발생했을 때 모두 호출되는 어드바이스
- ③ @Around : 타깃 객체의 메서드가 호출되는 전 과정을 모두 담을 수 있는 가장 강력한 어드바이스
- ④ @AfterReturning : 타깃 객체의 메서드가 정상적으로 실행을 마친 후에 호출되는 어드바이스

정답: ①

해설: @Before 는 메서드가 실행되기 전에 호출된다.

[문제 18] 어노테이션에 의한 트랜잭션은 @Transactional 어노테이션으로 설정가능하다. 다음 중 설정파일에 반드시 추가해 줘야 하는 태그는?

- ① < tx:annotation-driven ../ >
- ② < aop:transactional ../ >
- ③ < tx:transactional ../ >
- ④ < aop:annotation-driven ../ >

정답: ①

해설: @Transactional 을 활성화하려면 [tx:annotation-driven](#) 을 반드시 추가해야 한다.

[문제 19] MyBatis 에서의 예외처리 방식으로 틀린 것은?

- ① JDBC 코드는 SQLException 을 주로 발생시킨다.
- ② Spring-MyBatis 는 SQLException 을 DataAccessException 으로 re-throwing 한다.
- ③ 대부분의 SQLException 은 복구가 불가능하다.
- ④ DataAccessException 은 Checked Exception 이다.

정답: ④

해설: DataAccessException 은 런타임 예외(unchecked 예외)이다.

[문제 20] 인터셉터에 대한 설명 중 틀린 것은?

- ① HandlerInterceptorAdapter 를 상속받아 작성한다,
- ② preHandle 메소드를 구현하여 요청 이전에 호출을 가로챈다.
- ③ postHandle 메소드를 구현하여 뷰를 실행한 후에 호출을 가로챈다.
- ④ 로그인 처리나 접근내용 기록을 할 때 사용할 수 있다.

정답: ③

해설: postHandle 은 컨트롤러가 실행된 후, 뷰를 렌더링하기 전에 호출된다.

[문제 21] SqlSessionFactoryBean 에서 설정되는 내용이 아닌 것은?

- ① mybatis 기본 설정 파일 설정
- ② mapper xml 파일 위치 설정
- ③ 데이터소스 설정
- ④ 데이터베이스 종류 설정

정답: ④

해설: 데이터베이스 종류는 DataSource 에서 관리한다.

[문제 22] 데이터소스에 대한 설명으로 관계없는 것은?

- ① 데이터베이스에 연결해서 Connection 확보를 목적으로 한다.
- ② 스프링에서는 DataSource 를 통해 커넥션을 가져온다.
- ③ 커넥션 풀을 사용하려면 DataSource 가 아닌 다른 기술을 사용해야 한다.
- ④ 프러퍼티 파일을 사용해 변경되는 내용은 별도로 작성이 가능하다,

정답: ③

해설: DataSource 를 통해 커넥션 풀을 구성할 수 있다.

[문제 23] 스프링에서 파일 업로드 시 업로드 된 파일을 다루기 위한 클래스명으로 적합한 것은?

- ① MultiPartFormData
- ② MultipartUpload
- ③ MultipartFile
- ④ MultipartModel

정답: ③

해설: 스프링에서는 MultipartFile 을 사용해 업로드된 파일을 처리한다.

[문제 24] RESTful 서비스에서 사용되는 HTTP Method 가 아닌 것은?

- ① GET
- ② POST
- ③ DELETE
- ④ ADD

정답: ④

해설: RESTful 표준 메서드는 GET, POST, PUT, DELETE 등이 있으며, ADD 는 없다.

[문제 25] HTTP 요청 파라미터를 처리할 수 있는 어노테이션은?

- ① @PathVariable
- ② @RequestMapping
- ③ @RequestParam
- ④ @Autowired

정답: ③

해설: @RequestParam 은 요청 파라미터를 처리하기 위한 어노테이션이다.

[문제 26] Controller 에서 사용되는 어노테이션이 아닌 것은?

- ① @Controller
- ② @Aspect
- ③ @RequestMapping
- ④ @Autowired

정답: ②

해설: @Aspect 는 AOP 용 어노테이션으로 컨트롤러와 직접 관련은 없다.

[문제 27] 스프링 MVC 의 특징 중 틀린 것은?

- ① 서블릿 기반의 웹개발을 지원한다.
- ② Front Controller 패턴을 프레임워크 차원에서 지원한다.
- ③ 스프링을 기반으로 하기 때문에 스프링의 트랜잭션 기능이나 DI 기능을 사용할 수 있다.
- ④ 서버로 들어오는 모든 요청을 GenericController 가 선처리 한다.

정답: ④

해설: 스프링 MVC 에서 요청을 선처리하는 것은 DispatcherServlet 이다.

[문제 28] Layered Architecture 의 대표적인 사례로 MVC 패턴이 있다. 다음 중 MVC 패턴을 구성하는 요소 중 관계 없는 것은?

- ① Model
- ② View
- ③ Code
- ④ Controller

정답: ③

해설: MVC 는 Model, View, Controller 로 구성되며, Code 는 관계없다.

[문제 29] 스프링 컨테이너와 같은 의미로 쓰이는 용어가 아닌 것은?

- ① Bean Container
- ② Application context
- ③ Bean Factory
- ④ Dependency Injection

정답: ④

해설: Dependency Injection 은 IoC 컨테이너의 기능이지만, 컨테이너 자체를 의미하지 않는다.

[문제 30] DAO 디자인 패턴에 대한 설명중 잘못된 것은?

- ① 트랜잭션을 주로 담당한다.
- ② 비즈니스 서비스와 로우레벨 데이터 액세스 작업을 분리한다
- ③ 데이터베이스와 비즈니스 레이어 가운데 위치한다.
- ④ CRUD 작업을 수행한다.

정답: ①

해설: 트랜잭션 관리는 주로 서비스 레이어에서 담당한다.

[문제 31] DI 에 대한 설명 중 옳은 것은?

- ① Dependency Integration 이라고 함
- ② 컨테이너에 있는 빈을 찾는것
- ③ 클래스의 의존관계를 컨테이너가 자동으로 연결해 주는 것
- ④ 저장소에 저장된 빈에 접근하는 것

정답: ③

해설: DI(Dependency Injection)는 의존관계를 외부에서 주입받아 연결하는 개념이다.

[문제 32] JSON 에 설명중 잘못된 것은?

- ① 경량 데이터 교환 형식이다.
- ② JSON 표현식은 용량이 작고 기계가 이해하고 쉬우나 사람은 이해하기 어렵다.
- ③ 특정 언어에 종속되지 않으며, 대부분 언어에서 JSON 처리 라이브러리를 제공한다.
- ④ 키/값 형식의 구성을 따른다.

정답: ②

해설: JSON 은 기계와 사람이 모두 이해하기 쉽다.

[문제 33] 메이븐에서 기본적으로 설정된 소스 디렉토리 구조가 아닌 것은?

- ① /src/main/java
- ② /src/main/resources
- ③ /src/test/java
- ④ /src/server/java

정답: ④

해설: Maven 의 표준 디렉토리 구조에는 /src/server/java 가 없다.

[문제 34] Convention over Configuration 을 설명한 것 중 잘못된 것은?

- ① 설정보다는 관행에 따라 미리 정해진 설정으로 개발을 수행하는 것을 선호한다는 의미이다.
- ② 널리 알려진 best practice 에 따라 디렉토리 구조나 소스코드 일부를 생성해준다.
- ③ 메이븐이 이 방식에 따라 만들어 졌다.
- ④ 라이브러리에 대한 의존성 관리를 별도로 해준다.

정답: ④

해설: 의존성 관리는 Maven 의 기능이며, Convention over Configuration 과는 별개이다.

[문제 35] 다음 중 DI 의 유형이 아닌 것은?

- ① setter Injection
- ② Static Injection
- ③ Method Injection
- ④ Constructor Injection

정답: ②

해설: Static Injection 은 DI 방식으로 사용되지 않는다.

[문제 36] MVC 패턴에서 모델(Model)의 역할을 설명한 것 중 틀린 것은?

- ① 데이터 저장소와 연동
- ② 트랜잭션 처리
- ③ DAO, Service 클래스에 해당
- ④ View 와 컨트롤러의 가운데서 서로 연결해 주는 역할 수행

정답: ④

해설: 컨트롤러가 뷰와 모델 사이를 연결한다.

[문제 37] Spring MVC Framework 에서 파라미터를 전달받는 방식은 총 4 가지가 있다. 아래 방법 중 틀린 것은 무엇인가?

- ① @PathVariable
- ② HttpServletRequest
- ③ @RequestParam
- ④ ModelAndView

정답: ④

해설: ModelAndView 는 반환값이며, 요청 파라미터 전달 방식은 아니다.

[문제 38] 다음 중 요청파라미터의 값이 옵션이고, 파라미터의 값이 없을 때는 0 으로 기본값이 되는 문법으로 적합한 것은 무엇인가?

- ① (@RequestParam(value="author" required=true, defaultValue=0) String author)
- ② (@RequestParam(value="author" required=false, default=0) String author)
- ③ (@RequestParam(value="author" required=false, defaultValue=0) String author)
- ④ (@RequestParam(key="author" required=false, defaultValue=0) String author)

정답: ③

해설: required=false 와 defaultValue 를 같이 사용한다.

[문제 39] 다음 중 게시판을 RESTful 하게 개발하려고 URI 설계를 할 때 가장 적합하지 않는 설계를 고르세요.

- ① 글등록 : POST 방식 /bbs
- ② 전체글보기 : GET 방식 /bbs
- ③ 글삭제 : POST 방식 /bbs/delete/{articleId}
- ④ 글수정 : PUT 방식 /bbs/{articleId}

정답: ③

해설: RESTful 에서 삭제는 DELETE 메서드를 사용해야 한다.

[문제 40] 다음 중 RESTful 설계 원칙에 맞지 않는 URL 경로를 고르세요.

- ① GET /products/{productId} - 상품 상세 조회
- ② DELETE /products/{productId} - 상품 삭제
- ③ POST /products/update/{productId} - 상품 정보 수정
- ④ POST /products - 상품 등록

정답: ③

해설: 수정은 PUT/PATCH 를 사용하고, URI 에 update 라는 동사를 넣지 않는다