자바실무프로젝트 중간고사 정리

1. 2주차 퀴즈
2. 서블릿 클래스가 꼭 무조건 상속받아야하는 클래스는?
3. HttpServlet
4. 서버의 WEB-INF 폴더는 클라이언트가 접근할 수 없는 폴더이다.
5. O
6. 특정 Servlet 요청 전체 경로의 URL을 요청하면 복잡하고 보안에도 취약하기 때문에 간단하게 경로를 표현하고자 설정하는 것을 무엇이라고 하는가?
7. 서블릿 매핑(mapping)
8. 위 3번의 작업과 관련된 xml 태그가 아닌것은?
9. <servlet-name>
10. <servlet-class>
11. <name>
12. <url-pattern>
13. 3주차 퀴즈
14. 서블릿 URL매핑 설정 방식 2가지는?
15. 어노테이션 방식
16. xml 방식
17. 위 1번 매핑 설정방식과 관련된 코드는?
18. @Servlet
19. @WebProgram
20. @Server
21. @WebServlet
22. 서블릿 클래스에서 서버side의 정보저장을 위한 application객체를 얻는 메소드는?
23. request.getServletContext()
24. request.getSession은 HttpSession객체 리턴
25. Cookie는 어디 side에 정보를 저장하는 방식?
26. Client Side
27. 정보가 저장된 Cookie를 Client에 저장하기 위해 사용하는 객체는?
28. response객체
29. 4주차 퀴즈
30. 서블릿 클래스의 메소드 service(), doGet(), doPost() 중에 가장 먼저 호출되는 콜백메소드는?
31. service()
32. MVC모델에서 Controller의 결과를 View Jsp페이지로 연결해주는 객체 종류는?
33. Dispatcher(RequestDispatcher)
34. Jsp의 표현식으로 <%= request.getAttribute("result") %>를 표현언어로 적으면?
35. ${ result }
36. 서블릿 필터 클래스를 사용하는 목적은? 또 서블릿 필터 클래스를 등록해 두는 방법은?
37. 기능을 호출 전/후에 실행시키고 싶을 때
38. web.xml에 등록하거나 어노테이션 방식으로 설정
39. 서블릿 객체가 만들어지는 시점은?
40. 클라이언트의 요청이 있을때
41. 서블릿 객체의 라이프 사이클을 관리하는 것은?
42. 서블릿 컨테이너에 의해서
43. 서블릿(객체) 이란 무엇인가?
44. 클라이언트의 요청 처리를 하기 위한 서버의 worker
45. 5주차 퀴즈
46. 스프링 프레임워크의 장점?
47. 빠른 구현 시간
48. 쉬운 관리
49. 개발자의 역량 획일화
50. 검증된 아키텍처의 재사용과 일관성 유지
51. 스프링 프레임워크의 특징?
52. 경량(POJO - Plain Old Java Object)
53. IoC(Inversion of Control, DI 지원)
54. 관점지향프로그래밍(AOP - Aspect Oriented Programming) 지원
55. 컨테이너 역할
56. 결합도란 무엇이며, 결합도를 낮추기 위한 방법은?
57. 결합도는 하나의 클래스가 다른 클래스와 얼마나 많이 연결되어 있는 지를 나타내는 것
58. 다형성과 디자인 패턴(Factory) 이용
59. 6주차 퀴즈
60. 스프링 프로젝트의 라이브러리 build 관리툴은 무엇이며, 그 관리툴의 빌드의 여러가지 정보를 담고있는 파일은?
61. Maven(메이븐)
62. pom.xml
63. 스프링 컨테이너에 관리되어지는 자바 객체를 무엇이라고 부르는가?
64. Bean(빈)
65. 스프링 컨테이너의 로딩 순서는?
66. 컨테이너 생성(ApplicationContext 인터페이스를 구현한 GenericXmlApplicationContext 클래스를 통해)
67. Bean들이 들어있는 XML 설정 파일 읽음(어노테이션의 방법도 있음)
68. XML 파일에 등록된 Bean들의 Life Cycle과 Dependency가 관리되기 시작

1. bean 객체의 scope 속성 기본값은?
2. Single톤
3. 7주차 퀴즈
4. IoC와 DI가 어떤 말의 줄임말이며, 무엇인지 간단히 설명하시오.
5. IoC
6. Inversion of Control
7. 프로그램의 흐름을 개발자가 아닌 프레임워크가 주도
8. DI
9. Dependency Injection
10. 컨테이너가 의존관계를 bean설정 정보를 바탕으로 자동으로 연결
11. DI 구현 방법 2가지는?
12. setter injection
13. constructor injection
14. <bean id="tv" class="polymorphism.SamsungTV">을 자바코드로 표현한다면?
15. SamsungTV tv = new SamsungTV();