**서 블 릿**

|  |  |
| --- | --- |
| 1 | 다음 중 틀린 것은?  (1) 웹서버에서 실행되는 javaEE 기반의 클래스를 서블릿이라 한다.O  (2)서블릿을 이용하여 HTML 컨텐츠를 생성하려면 out.print() 메서드 안에 문자열로 넣어야 하므로 개발 효율성이 떨어진다.O  (3)디자이너는 자바의 코드와 문법을 알 수 없으므로 서블릿으로 HTML 컨텐츠를 출력하는 코드는 협업에  있어 문제가 많다. O  (4)서블릿으로 HTML 컨텐츠를 출력하는 방식이 효율성이 떨어지므로 이를 개선하기 위해 개발된 스크립트가 바로 JSP이다.O  (5)JSP는 자바 코드 안에 HTML을 넣으려는 것이 아니라, HTML 코드 안에 자바 코드를 넣을 수 있는 방법을 제공한다.O  (6)많은 문제로 인하여 더 이상 서블릿은 사용되지 않는다.X(서블릿은 피해갈 수 없다.) |
| 2 | 다음 설명 중 틀린 것은?  (1) HTML 태그와 혼용 시 소스가 복잡해지는 서블릿의 단점은 JSP로 보완 되어질 수 있다.O  (2) JSP는 서블릿으로 변환되어 실행된다.따라서 JSP가 곧 서블릿이다.O  (3) 서블릿은 컨테이너에 의해서만 자동으로 생성되어지며 프로그래머가 작성할 수는 없다.X  (4) JSP의 <%! %> (선언부)영역에 코딩을 하면 서블릿의 멤버 영역에 코딩한 것과 같다.O  (5) JSP의 <% %> 영역에 코딩을 하면 서블릿의 service() 메서드에 코딩한 것과 같다.O |
| 3 | 다음 설명 중 틀린 것은?  (1)서블릿으로서 반드시 가져야 할 가장 기본이 되는 메서드만을 정의해 놓은 최상위 객체는 Servlet 인터  페이스이다.O  (2)HttpServlet 클래스는 GenericServlet 클래스를 상속 받으며, HTTP 기반의 요청을 처리하기에 최적화 된  서블릿이다.O  (3)HttpServlet 이 HTTP 프로토콜이라는 특정 프로토콜 상에서의 서블릿을 정의한 것이라면 GenericServlet은 일반적인 프로토콜에서의 서블릿의 업무 내용을 규정한 추상 클래스이다.O  (4)GenericServlet은 Servlet 인터페이스를 구현하고 있으며, HttpServlet 은 GenericServlet 을 상속받고 있다O  (5)클라이언트의 요청을 처리하는 doXXX 형 메서드는 Servlet 인터페이스에서 명시되어 있다.X  (6)Sun에서 서블릿을 추상클래스로 선언한 이유는 , 실제 구현 의무를 서버 제작사에게 부여하기 위함이다.O |
| 4 | HttpServlet 클래스의 메서드에 대한 설명 중 틀린 것은?  (1)클라이언트의 요청이 Get 방식일 경우 doGet() 메서드가, Post 방식일 경우엔 doPost() 가호출된다.O  (2) service() 메서드는 클라이언트의 요청 방식에 따라 doGet() 혹은 doPost() 메서드에 의해 호출되어 진다.O  (3) doGet() 과 doPost() 등의 메서드는 service() 메서드에 의해 호출되어 진다.O  (4) service() 메서드의 실행은 쓰레드가 담당한다.O  (5) 서블릿 인스턴스가 메모리에 올라가는 시점은 컴파일 후, 최초로 클라이언트의 요청이 있을 때이다.O |
| 5 | 다음 설명 중 맞는 것은?  (1)서블릿의 인스턴스는 프로그래머가 new 연산자를 통하여 메모리에 올려야 한다.X  (2)서블릿의 인스턴스는 클라이언트의 요청이 있을 때마다 메모리에 생성되어진다.X  (3)클라이언트의 요청이 있을 때마다 서버측 에서는 쓰레드가 생성되며, 이 쓰레드에 의해 서블릿의 service()메서드가 실행되어 진다.O  (4)클라이언트의 요청에 대한 처리가 끝나면 서블릿은 메모리에서 소멸된다.X(서버가 종료될 때 소멸) |
| 6 | 다음 설명 중 맞는 것은?  (1)서블릿의 생명 주기 메서드 중 가장 먼저 호출되어 지는 메서드는 init() 메서드이며, 클라이언트의 요청이있을 때마다 호출되어 진다.X  (2)init()메서드는 서블릿의 초기화를 담당하므로, 서블릿의 생성자로 볼 수 있다.X(생성자보다 느리다.생성자랑은 다른 메서드이다.)  (3)서블릿은 컨테이너가 생성하므로, 생성자가 존재할 수 없다.x(디폴트생성자가 존재한다)  (4)서블릿의 생명 주기와 관련된 메서드에는 init(), service(), destroy() 가 있다.O |
| 7 | 다음 설명 중 틀린 것은?  (1)서블릿 객체가 생성되어 질때, 자신이 속한 웹어플리케이션의 설정 정보를 컨테이너로부터 부여 받게  되는데 이 정보를 가진 객체가 바로 ServletConfig 인터페이스이다.  (2) 서블릿 객체 생성시, 생성자 보다 앞서 호출되는 메서드가 바로 init() 메서드이다.  (3) 어플리케이션의 정보를 담는 ServletConfig 객체는 init() 메서드의 인수로 전달되어 진다.  (4) 초기화 정보 중엔 web.xml 파일에 설정된 파라미터 값 정보도 포함된다. |
| 8 | 서블릿에 대한 설명 중 맞는 것은?  (1)HttpServletRequest 는 JSP 스크립트에서 사용하였던 request 내장 객체의 데이터 타입이다.O  (2)HttpServletResponse 는 JSP 스크립트에서 사용하였던 response 내장 객체의 데이터 타입이다.O  (3)HttpServletRequest 객체는 인터페이스이므로 프로그래머가 new 할 수 없으며 컨테이너에 의해 메모리에올라간다.O  (4)클라이언트가 브라우져를 5번 새로고침을 눌렀다면 HttpServletRequest 객체는 메모리에 5번  생성되어 진다.O |
| 9 | 다음 설명 중 맞는 것을 고르면?  (1)서블릿을 프로그래머가 작성하려면 Servlet 이나 HttpServlet 을 상속받으면 된다.O  (2)서블릿의 초기화,소멸 등의 생명 주기는 컨테이너가 관리하며 생명 주기와 관련된 메서드에는 doGet, doPost 등이 있다.X(생명주기는 init(),service(),destroy())  (3)init()은 서블릿의 생성자이다.X  (4)서블릿에는 생성자를 두면 안된다.X(둬도 된다.)  (5)서블릿의 초기화시 DD(Deployment descriptior-배포서술자=web.xml)로부터 각종 설정 정보를 가져올 수 있으며, 이 역할은 init() 메서드가 담당한다.O |
| 10 | 다음 설명 중 맞는 것은?  (1)서블릿의 메서드중 init(),service(),destroy() 를 생명주기 메서드라고 하며, HttpServlet 클래스에만 정의되어 있다.X(서블릿 인터페이스에 있다)  (2)init() 메서드는 서블릿의 일생에 단 한번만 호출되어 진다.O  (3)init() 메서드는 servlce() 메서드가 호출 되고 난 직후에 컨테이너에 의해 호출되어 진다.X  (4)생명주기 메서드를 호출하는 주체는 서블릿이다.X(주체는 웹 컨테이너이다.) |
| 11 | 다음 설명 중 맞는 것은?  (1)클라이언트의 요청이 들어오면 컨테이너가 service() 메서드를 호출하므로 비즈니스로직을 service()  메서드에 재정의해야 한다.X(doxx메서드를 재정의)  (2)쓰레드는 요청마다 하나씩 할당되어지는 것이 아니라, 클라이언트당 하나씩 할당되어 진다.따라서 같은  클라이언트가 여러 번 요청을 시도하면 쓰레드는 하나만 할당되어 진다.X  (3) Request 와 Response 객체는 클라이언트당 하나씩 할당되어 지므로, 같은 클라이언트가 브라우져를  여러 번 새로고침을 눌러도 Request , Response 객체는 여러 번 생성되지 않는다.X(들어갈때마다 생성된다.즉, 소모품과 같다)  (4) HTTP 메서드에는 GET, POST 방식 이외에도 여러 가지가 있지만, GET ,POST 이외에는 거의 사용할  일이 없다.O(REST가 유행이므로, 그 이외의 방식도 사용하고 있다.) |
| 12 | 서블릿에 대한 설명 중 맞는 것은?  (1)클라이언트의 요청에 응답하는 서블릿은 매 요청시마다 서블릿 객체를 새로 생성시킨다.X  (2)A와 B라는 두 클라이언트가 같은 서블릿을 요청할 경우, 서블릿 객체는 A,와 B에 대해 각각 생성되어 요청을 처리한다.X  (3)JSP 소스의 변경에 의해 서블릿이 재 컴파일 되는 시점은 ,JSP 소스의 변경이 있은 후로 처음 요청을 시도하는 클라이언트에 의해서이다.O  (4)JSP에 의해 생성된 서블릿은 개발자가 정의한 서블릿과 다를바가 없으므로, 서블릿을 재정의할때 그대로 사용해도 된다.X(웹 컨테이너가 만든 것과 살짝 다르다) |
| 13 | 다음 설명 중 틀린 것은?  (1)클라이언트의 “요청” 에 대한 개념을 HttpServletRequest 인터페이스로 정의하며,이 인스턴스를 request객체라 한다.O  (2)HttpServletRequest 인터페이스에 대한 API 문서는 Sun사에서 제공한다. O  (3)HttpServletRequest 인터페이스의 인스턴스인 request 객체는 컨테이너에 의해 메모리에 올라간다.O  (4)클라이언트와 서버의 개념은 HTTP 프로토콜에서만 사용되는 개념이다.X(이 개념은 언제나 웹용이 아니다.) |
| 15 | 다음 설명 중 틀린 것은?  (1)컨테이너는 클라이언트의 요청이 들어오면, 요청을 담당할 서블릿 객체의 service() 메서드를 호출한다.O  (2) 서블릿의 service() 메서드는 내부적으로, 클라이언트의 요청 방식에 따라 doXXX 형 메서드를 호출하도록 되어 있다.O  (3)프로그래머가 서블릿을 프로그래밍 할때는 service()메서드 보다는 doXXXX() 메서드들에 비즈니스 로직을 구현하는 것이 대부분이다.O  (4)컨테이너는 클라이언트의 요청이 들어오면, request 객체만을 생성하여 서블릿에게 전달할 뿐, response객체는 요청 단계에서는 필요가 없으므로 전달하지 않는다.X  (5)클라이언트의 요청이 있을 때마다 매번 쓰레드가 생성되어 서블릿의 service() 메서드를 수행하고 이  service() 메서드를 수행하고 나면 쓰레드는 소멸한다.O |