1. 객체와 생성자

- 객체: 데이터와 로직을 묶어놓은 일종의 추상적 틀인 클래스로부터 만들어진 복사본  
 - 생성자: 객체 생성시 객체를 초기화하고 heap에 저장하는 메소드, 생성자는 클래스명과 동일하며 종료할 때는 값을 반환하지 않는다.

a. New 연산자가 객체가 저장될 메모리 공간 할당  
b. 생성자가 Image 객체를 초기화   
c. new 연산자가 새로 생성된 객체의 주소를 변수에 저장  
d. 변수 명을 통해 객체에 접근 가능

2. Annotation

비즈니스 로직에 영향을 주지는 않지만 해당 타겟의 연결 방법이나 소스코드의 구조를 변경할 수 있음

3. CGI(Common Gateway Interface)

- 웹 서버와 외부 프로그램 사이에서 정보를 주고받는 방법이나 규약

- 사용자의 요청에 의한 서버의 응답

- C, Perl, Visual Basic

- PHP / ASP는 CGI를 만들기 위한 한가지 도구이며 한가지 스크립팅 언어

- 서버측에서 수행되는 다른 응용프로그램과 통신할 수 있는 기능을 제공

- CGI는 C, JAVA등의 언어로 사용하여 작성하게 되는데, 자바 CGI의 성능을 개선한 방법을 Servlet

4. Servlet

- Tomcat위에서 동작하는 java 프로그램

- 서버에서 실행되는 작은 프로그램

- WAS를 통해 컴파일 된 후 메모리에 적재되어 클라이언트의 HTTP Get, Post 등의 요청을 처리하는 자바 프로그램

- 자바를 이용해서 작성한 CGI프로그램

- 일반 CGI는 특성상 클라이언트의 요청이 있을 때마다 프로세스를 생성하여 CGI를 수행하지만클라이언트의 요청이 많아질 경우 과부하를 불러오기 때문에 비효율적

- 이러한 단점을 보안한 것이 Servlet이다. Servlet은 클라이언트의 요청이 생길 경우 프로세스 대신 Thread를 생성하고 각 Servlet은 한번만 메모리에 로드되고 계속 사용된다. Servlet을 실행시키는 JVM은 항상 실행되고 있는 상태이고, 사용자가 많은 사이트의 경우 효과적

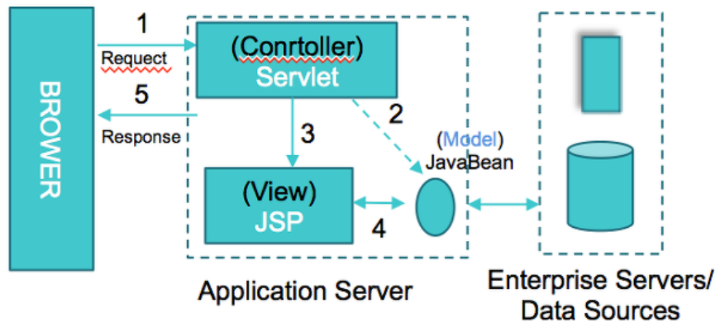
- Servlet을 만들기 위해서는 자바 코드를 작성하고, 코드를 컴파일 해서 클래스 파일을 만들어야 한다. 즉 Servlet은 실행 코드 방식에 속한다. 따라서 Servlet을 이용한 웹 어플리케이션을 개발할 경우 화면에 출력되는 데이터를 조금만 바꾸고 싶어도 코드를 수정하고 컴파일하는 작업을 반복해 주어야한다.

때문에 Sun에서 Servlet의 단점을 보완하기 위해서 스크립트 방식의 표준인 JSP를 만든 것이다.

5. JSP

- Servlet의 사용을 통해 웹페이지의 내용이나 모양을 제어하는 기술

- Servlet은 JAVA안에 HTML태그가 섞여있어 작업에 대한 분리적 측면에서 효율성이 떨어졌고 이러한 불편을 해결하기 위해 JSP개발

 - HTML내부에 Java코드가 있어 HTML코드를 작성하기 유리함

6. Process vs Thread

- Process: 운영체제로부터 자원을 할당 받은 작업의 단위, 실행중인 프로그램에 대한 인스턴스

- Thread: 프로세스가 할당 받은 자원을 이용하는 실행의 단위

- Process간 통신보다 Thread간 통신의 비용이 전역 변수를 이용하여 적음, 동기화 문제 주의

- Thread는 같은 Process내의 주소 공간, 메모리 등의 자원을 공유하며 실행

Java Reflection

* 객체를 통해 클래스의 정보를 분석해 내는 프로그램 기법

BeanFactory

* 어플리케이션이 실행한 후 객체가 호출 될 당시 객체의 인스턴스를 생성

GetAsString() vs toString()

* GetAsString(): convenience method to get this element as a string value
* ToString(): Returns a String representation of this element

TLS

URL vs URL

URL: Uniform Resource Locator: 리소스 위치 표시

URI: Uniform Resource Identifier, URI = URL+URN

URN: Uniform Resource Name

Storage Engine

데이터 베이스 엔진(Database engine)

데이터베이스에 대해 데이터를 삽입, 추출, 업데이트 및 삭제하는데 사용하는 기본 소프트웨어 컴포넌트

MYSQL의 경우 실제 데이터를 디스크 스토리지에 저장하거나 디스크 스토리지로부터 데이터를 읽어오는 부분에서 DBMS 고유의 사용자 인터페이스를 이용하는 방법과 포트번호를 통해 하는 방법 제공

대부분의 관리시스템은 DBMS의 사용자 인터페이스가 아닌 내장된 엔진과 상호작용을 할 수 있는 자신만의(API)를 포함하고 있으며 데이터 베이스 엔진이라는 용어는 종종 데이터베이스 서버 또는 데이터베이스 관리 시스템이라는 용어로 사용되기도 하며 대부분 다중 엔진 지원