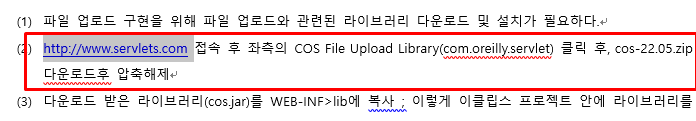
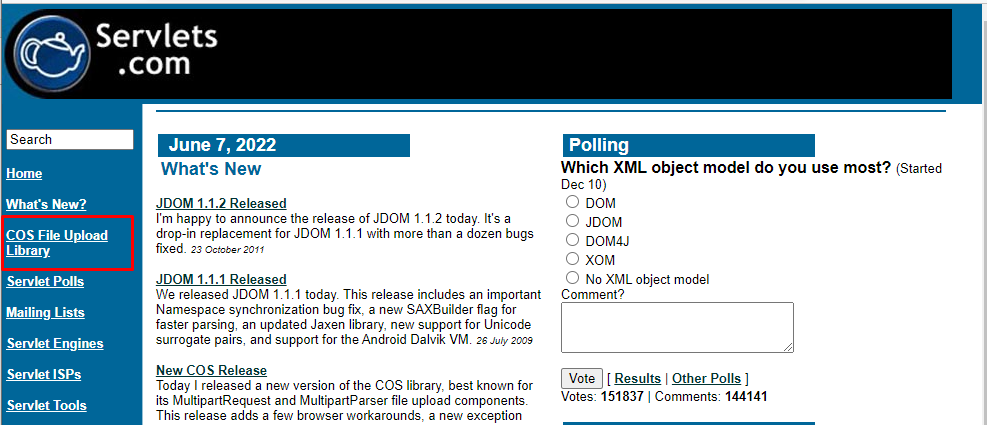
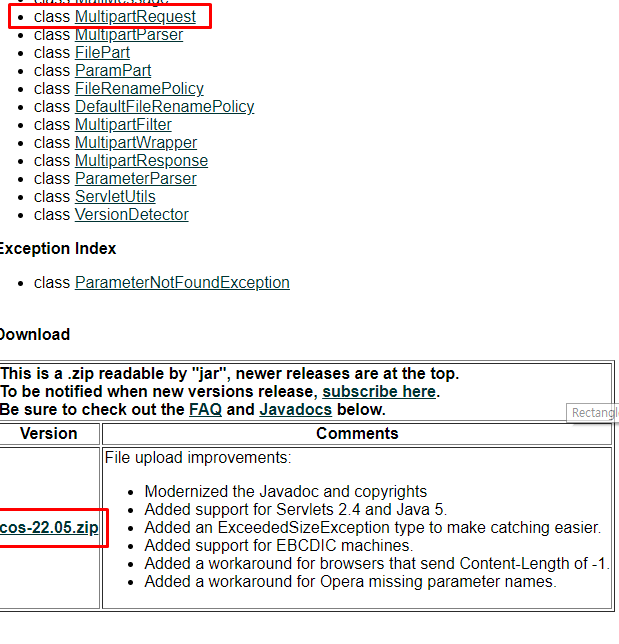
[ 14 ] 파일업로드



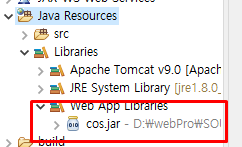




파일 업로드 라이브러리 설치

* 1. 파일 업로드 구현을 위해 파일 업로드와 관련된 라이브러리 다운로드 및 설치가 필요
  2. <http://www.servlets.com> 접속 후 좌측의 COS File Upload Library(com.oreilly.servlet) 클릭 후, cos-22.05.zip 다운로드후 압축해제
  3. wepPro의 다운로드 폴더에 옮겨놓기
* 어떨땐 파일 첨부를 할거고, 어떨땐 파일첨부를 하지 않을 것
* 모든 프로젝트에 파일첨부를 할거면 톰캣폴더에 복사해놓으면 되지만 우린 그렇게 안할거니까 굳이 이렇게 할 필요는 없음
* 다운받은 cos-22.05/lib/cos.jar를 이클립스의 WEB\_INF>lib에 복사 : 이 프로젝트 안에서만 쓸거야!(모든 프로젝트에서 쓸거면 톰캣라이브러리에 옮겨놓으면됨 )

이작업을 완료하면 자동으로 아래 그림처럼 생성이 됨

web App Libraries가 생김

**[lect1\_fileUpGrammar]**

- 구현은 lect1에서 할거고, 파일을 첨부한건 fileUpFolder에 저장시킬 것

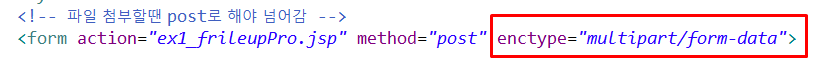
- jsp로 받아도 되고, <input>타입으로 form태그 활용한 내용만 들어갈 것이기 때문에 html파일로만들어도됨. 하지만 나중에 헤더와 풋터를 include 하려면 jsp로 만드는 것이 가장 나음

**\*ex1\_fileupForm.jsp**

: 파일을업로드할 form태그를 만드는 영역

1. form 태그를 만들 때 무조건 post 방식으로 만들어야함. 파일을 올릴 때 get으로 해버리면 파일 첨부가 안됨

**2. 파일을 올렸다고 해서 무조건 서버에 집어 넣으면 안됨. 첫번째 파일이 날라가고 두번째 파일로 대체 됨 이럴땐 enctype 이라는걸 써서 파일이겹치지 않도록 처리 해주어야함 (아래그림)**



3. - input 타입에 파일넣고, 파일 업로드 폼 만들기 (아래처럼)

--<form action=*"ex1\_frileupPro.jsp"* method=*"post"* enctype=*"multipart/form-data"*>

<p>파일<input type=*"file"* name=*"file"*></p>

<input type=*"submit"* value=*"파일업로드"* class=*"btn"*>

</form>

4. 데이터베이스에는 파일 이름만 저장, 서버에만 파일을 올려놓음 (가지고 오는데 너무 시간이 오래 걸 리기 때문에

\***ex1\_fileupPro.jsp**

1.fileUpFolder에 첨부한 파일을 업로드 하기 위해 path를 설정해주기

2.업로드 maxSize를 지정해주기 (1024\*1024 는 1기가) 우린 10기가로 지정

3.String filename =""; 서버에 저장될 파일 이름

4. 만약에 진영이가 h.jpg 파일을 업로드하고 다음날 내가 똑 같은 파일 이름을 써서 업로드를 하면 ?

-- 예린컴퓨터엔 h2.jpg로 자동으로 저장되게끔.하지만 예린 컴퓨터에 뜨는건 h.jpg로 뜰 수 있도록

-- originalFileName(업로드한 오리지널 파일이름) 과 filename(서버에 저장될 이름)을 따로 설정해두어야 함

5. MultipartRequest 객체생성해주기 //multipartRequest는 매개변수 없는 생성자 함수가 없음.

//매개변수 있는 생성자 함수 만들어 주면 됨

//하지만 얘는 try catch해줘야함. (자바에서 쓸거니까)

- 첫번째 매개변수에 request객체,

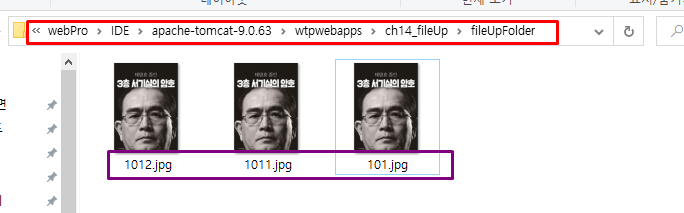
--두번쨰 매개변수에 save directory

--세번째 매개변수에 post 방식으로 들어왔을 때 maxSize

--네번째 매개변수에 한글이 꺠질수 있으니 인코딩 해주는 변수

--다섯번째 매개변수에 **new** DefaultFileRenamePolicy() 같은 이름의 파일이 있을 경우 rename을 자동 적으로 해주는 아이 : 1. jpg -> 11.jpg 이런식으로!

-6. 얘부터 실행해도 객체생성이 안됨 (아직 파일을 안받은 상황이니까!) fileUpform 부터 실행해줘야됨

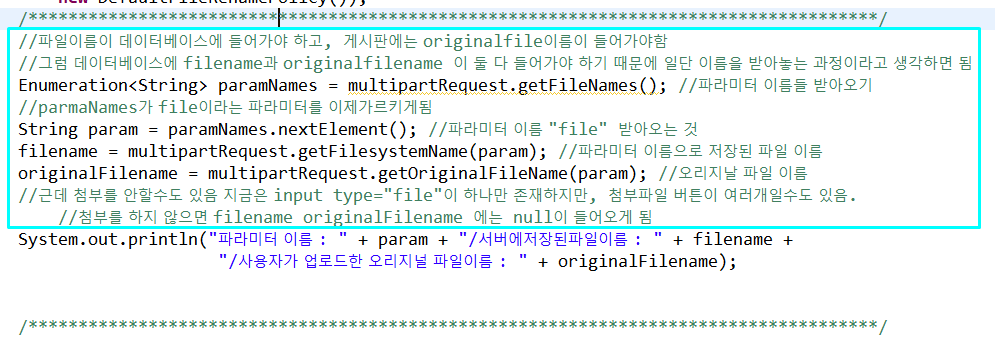


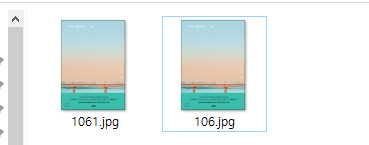
7. fileUpform을 실행시키고, 동일한 파일을 계속 업로드 하면 자동적으로 filename에 1이 붙어 생성이됨

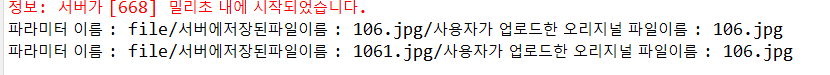
위그림의 빨간 박스 에 저장이 자동적으로 됨

8. 만약 내가 소스를 바꾸면 사진도 없어짐 하지만 데이터가 다른 곳에 복사 해놓음

9. 데이터베이스에 올릴 filename,과 originalname을 받아와야함 (아래 주석 및 이미지참고)







--고객에게 뿌릴건 : 오리지널파일이름

--서버에 저장시킬 건 : filename

-- 데이터 베이스에는 이 두개다 넣어야 하기 때문에 이름을 뽑는 과정이 필요

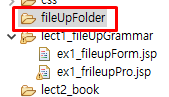
10. 서버에 업로드한 파일을 소스의 폴더로 file copy

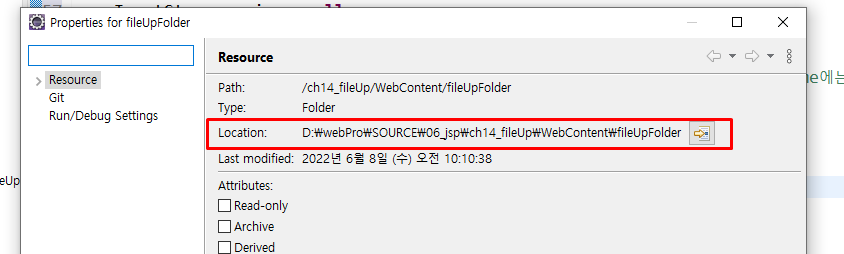
-- InputStream is = null; / OutputStream os = null;

-- try-catch활용해서 서버파일 열어주기

-- filename이 null이 아니면 (첨부를 했을경우) 복사를 할 것

-- 복사될 파일 (원본 ) 은 inputstream에 담고, 복사가 된 파일은 outstream에 담아주기

ㅍ 



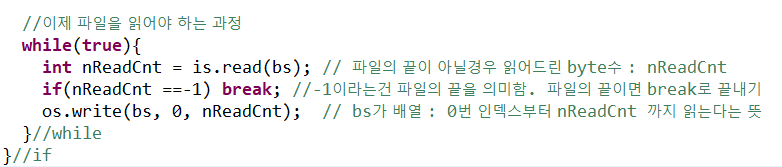


--빨간 박스의 경로를 os에 넣어주고, 슬래쉬 1개로 다 바꿔주던가, 역슬래쉬2개로 다 바꿔주기. 특히 마 지막 부분엔 꼭 닫아줘야 됨

-- 사진을 읽을 때 몇바이트씩 읽을지 저장해주기

ㄴ**byte**[] bs = **new** **byte**[(int)serverfile.length()]; // 서버파일의 크기만큼 바이트로 읽어내겠따 는 뜻

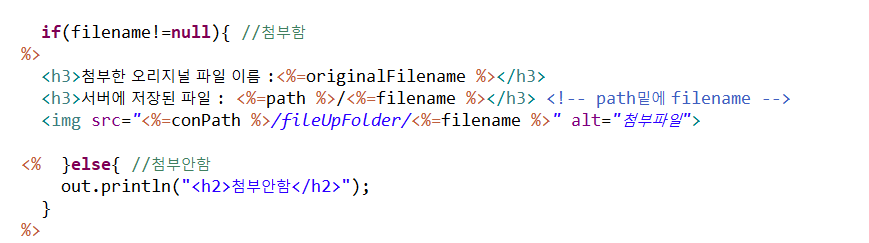
-- 이제 파일을 읽어야 하는 과정



11. 마지막으로 finally 로닫아주기

12. 내가 첨부를 했는지 안했는지 filename으로 알수 있음. filename이 null이 아니면 첨부를 한 것

근데 else로 빠지면 첨부안한 것



**\*ex2\_fileupForm3.jsp**

: 파일을업로드할 form태그를 만드는 영역 파일첨부 3개 만들 form

**\*ex2\_fileupPro3.jsp**

위와동일한 작업 거치면 됨.

1.fileUpFolder에 첨부한 파일을 업로드 하기 위해 path를 설정해주기

2.업로드 maxSize를 지정해주기 (1024\*1024 는 1기가) 우린 10기가로 지정

3.파일 업로드가 여러 개기 때문에 배열로 받아내기

--String[] filename={"","",""};

--String[] originalFilename={"","",""};

4. 만약에 진영이가 h.jpg 파일을 업로드하고 다음날 내가 똑 같은 파일 이름을 써서 업로드를 하면 ?

-- 예린컴퓨터엔 h2.jpg로 자동으로 저장되게끔.하지만 예린 컴퓨터에 뜨는건 h.jpg로 뜰 수 있도록

-- originalFileName(업로드한 오리지널 파일이름) 과 filename(서버에 저장될 이름)을 따로 설정해두어야 함

5. MultipartRequest 객체생성해주기 //multipartRequest는 매개변수 없는 생성자 함수가 없음.

//매개변수 있는 생성자 함수 만들어 주면 됨

//하지만 얘는 try catch해줘야함. (자바에서 쓸거니까)

- 첫번째 매개변수에 request객체,

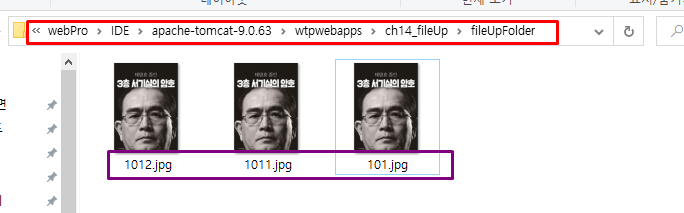
--두번쨰 매개변수에 save directory

--세번째 매개변수에 post 방식으로 들어왔을 때 maxSize

--네번째 매개변수에 한글이 꺠질수 있으니 인코딩 해주는 변수

--다섯번째 매개변수에 **new** DefaultFileRenamePolicy() 같은 이름의 파일이 있을 경우 rename을 자동 적으로 해주는 아이 : 1. jpg -> 11.jpg 이런식으로!

-6. 얘부터 실행해도 객체생성이 안됨 (아직 파일을 안받은 상황이니까!) fileUpform 부터 실행해줘야됨



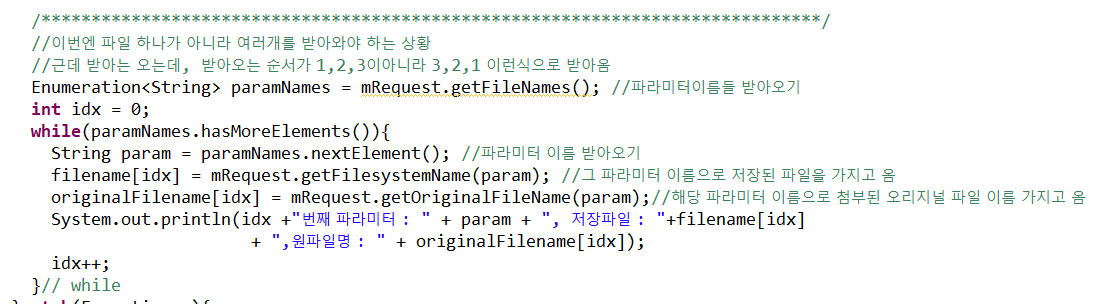
7. fileUpform을 실행시키고, 동일한 파일을 계속 업로드 하면 자동적으로 filename에 1이 붙어 생성이됨

위그림의 빨간 박스 에 저장이 자동적으로 됨

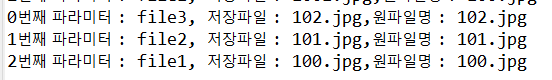
8. 만약 내가 소스를 바꾸면 사진도 없어짐 하지만 데이터가 다른 곳에 복사 해놓음

9. 데이터베이스에 올릴 filename,과 originalname을 받아와야함 (아래 주석 및 이미지참고)

-- 다만 이번엔 여러 개를 받아오는 상황







(순서가 반대로 들어오게 됨)

--고객에게 뿌릴건 : 오리지널파일이름

--서버에 저장시킬 건 : filename

-- 데이터 베이스에는 이 두개다 넣어야 하기 때문에 이름을 뽑는 과정이 필요

10. 서버에 업로드한 파일을 소스의 폴더로 file copy

-- 얘는 for문으로 돌려서 뽑아줘야함(몇 개인지 모르니까!)

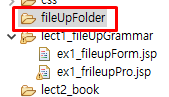
-- file이 num이 아닐때만 카피할거니깐 if문 안에다가 앞 로직 복붙

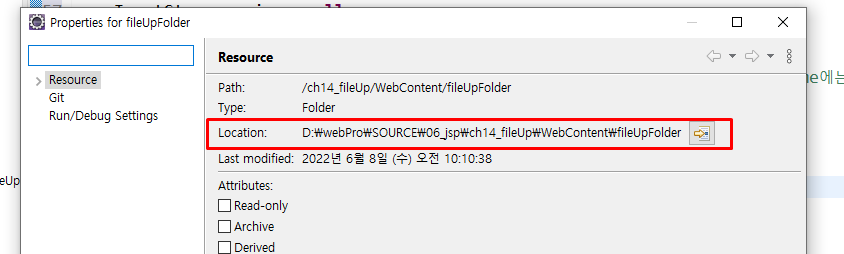
-- InputStream is = null; / OutputStream os = null;

-- try-catch활용해서 서버파일 열어주기

-- filename이 null이 아니면 (첨부를 했을경우) 복사를 할 것

-- 복사될 파일 (원본 ) 은 inputstream에 담고, 복사가 된 파일은 outstream에 담아주기

ㅍ 



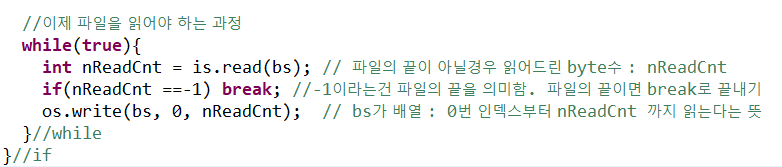


--빨간 박스의 경로를 os에 넣어주고, 슬래쉬 1개로 다 바꿔주던가, 역슬래쉬2개로 다 바꿔주기. 특히 마 지막 부분엔 꼭 닫아줘야 됨

-- 사진을 읽을 때 몇바이트씩 읽을지 저장해주기

ㄴ**byte**[] bs = **new** **byte**[(int)serverfile.length()]; // 서버파일의 크기만큼 바이트로 읽어내겠따 는 뜻

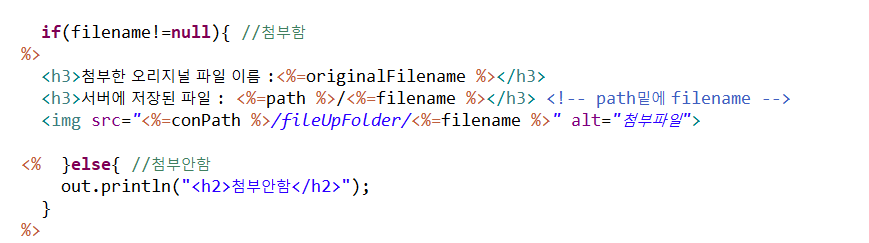
-- 이제 파일을 읽어야 하는 과정



11. 마지막으로 finally 로닫아주기

12. 내가 첨부를 했는지 안했는지 filename으로 알수 있음. filename이 null이 아니면 첨부를 한 것

근데 else로 빠지면 첨부안한 것



-----------------------------------06-09----------------------------------------

------------------------------------lec2\_book & bookimg-----------------------------------

책 리스트는 페이징 처리를 해줘야해

책을 누르면 책 상세보기 페이지로도 넘어가야해

실질적으로 첨부한 파일은 bookimg에, 파일에 대한 처리는 lect2\_book에 (대부분의 사람들이 구매후기에 사진 잘 안올리긴함) -- 책등록을 하는데 책 이미지를 첨부하지 않으면 default 이미지를 올려놓는 작업을 할 것

1. bookimg에 default 이미지 옮겨놓기 (noImg.png , NOTHING.JPG 옮겨놓기 )

2. 테이블 만들기

-- 책 이름 동일할수있음

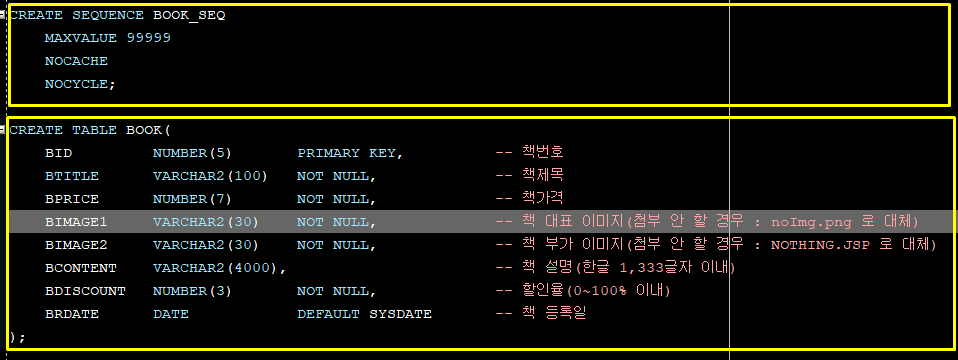
-- 이미지는 서버에 저장해놓고 파일 이름만 데이터베이스에 넣을 것

-- 프라이머리키로 만들게 없어서 SEQUENCE 만들어줘서 PRIMARY KEY 주기

-- 테이블 생성시 BLOB 쓰면 파일 자체를 넣을 순 있지만 파일은 안넣고 이름만 저장할거임. 안정적이지

도 않고, 가지고 오는데 시간이 오래걸림

-- 책 소개는 VARCHAR2(4000)을 쓸건데, 1333글자 사용 가능. 이것보다 길게 하고 싶으면 CLOB 해야함

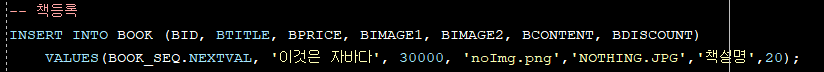


3. 책 등록하는 것, 책 뿌리는 것 , PAGING 처리, TOP-N구문 등 활용해서 쿼리 만들기

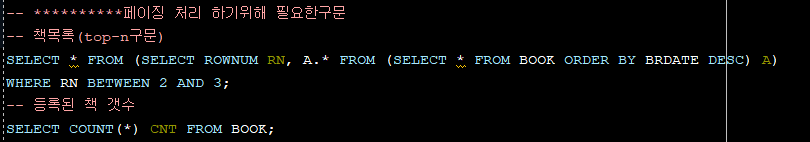
-- 책등록

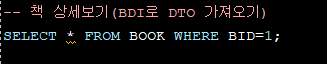
-- 책목록(책 LIST출력)

-- 페이징 처리 하기 위해 필요한 애들 (책목록-TOP-N구문, 등록된 책 개수)









4. DTO 만들기 (간혹가다 테이블이 여러 개일땐 DTO만드는게 좀 다를 수 있지만, 테이블이 하나이기 때문에 테이블 필드명과 동일하게 DTO 나오는것 )

-- 이번에는 빈 태그를 못 씀 (request 객체로 작업할게 아니라 file을 받기도 할거고, multipartrequest 객 체를 만들거기 때문에 빈을 활용을 못함.. 따라서 매개변수 없는 생성자 함수 굳이 만들 필요없음)

5. DAO 만들기

-- 책 등록 성공 유무에 따라 final static 변수 return 할 수 있도록 만들어주고

-- 싱글톤 객체 생성, 매개변수 없는 생성자 함수 만들고, 커넥션 풀 만들어 주기

-- 각각의 dao는 주석 확인하면서 직접 소스 체크하기

6. ex1\_bookRegisterForm.jsp 만들기

7. ex2\_bookRegisterPro.jsp 만들기

7-1) 파일첨부를 위한 과정

-- request로 "utf-8"을 설정해야함

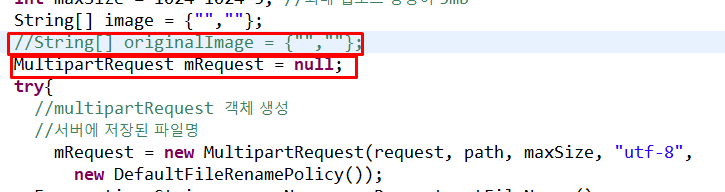
-- String path 설정하기

-- int maxSize 설정하기

-- String[] image ={"",""}; 해주기

- originalImage 를 여기선 안해도 됨. 게시판 같은 경우에는 내가 첨부한 파일 이름이 그대로 나왔으면 좋겠기 떄문에 그럴땐 originalFilename을 받아야 하지만, 책을 등록하는 거에서는 따로 original을 안넣어 도 됨

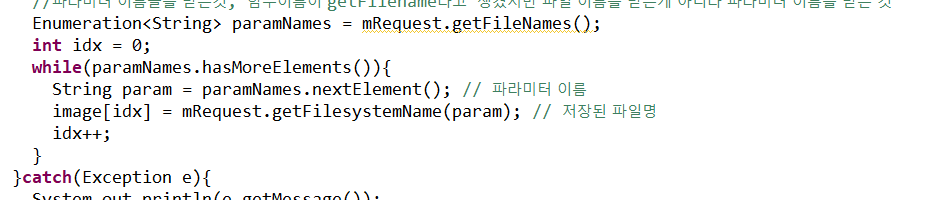
-- 여기에서는 catch절 아래에서 파일 첨부 외에도 파라미터에서 입력된 정보를 get할 때 multipartrequest가 쓰여야 하기 때문에 앞선 예제와는 다르게 변수를 try절 전에다 미리 빼줘야함



-- Enuveration<String> paramNames = mRequest.getFilNames(); 해주기

ㄴ-- 함수이름이 getFilNames 이라고 해서 파일 이름을받아오는게 아니라 파라미터들을 받아오는 것

--idx 를 초기화 시켜준 뒤 while문돌리기



-- catch절 나와서 서버(톰캣폴더)에 저장된 bookimg 폴더 파일을 (지금막첨부해서 추가된 파일들) 소스의 bookimg 폴더로 복사 (아래단)

-- for문으로 image뿌려주고 if(img!=null)일 경우 is,os 진행하기

-- inputStream, outputstream null값으로 세팅해주고, File serverFile = new File(patt+"/"+img); 해주기

-- try-catch 하나 더 만들고 try 절에 빨대 꼽고 빼기

-- is에 복사될 원본 파일 넣고(serverFile), os에 복사된 파일명 쓰기

-- 읽어드릴 byte 지정하기 위해 객체 , 배열 만들어주기

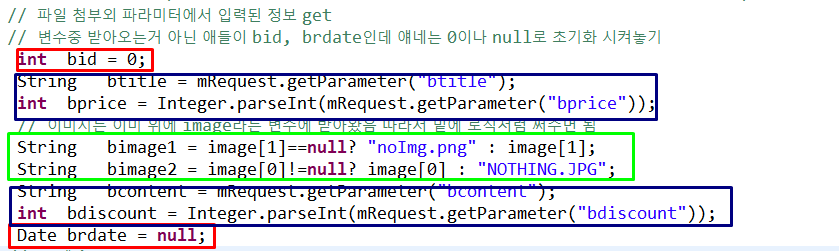
-- while(true)해서 돌려주고, int readbyteCnt = is.read(bs); 하고, readbyteCnt==-1이면 break로 빠져나오 고 os.write(bs, 0, readbyteCnt); bs 를 0번방부터 readbyteCnt까지 읽어줘라

7-2) 나머지 애들 받아오기

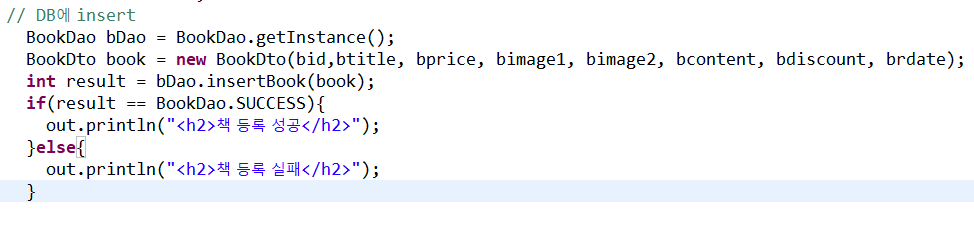
-- 변수중 받아오는게 아닌 아이들이 bid, brdate인데 얘네는 0이나 null로 초기화

-- bimage1, bimage2 은 이미 위에 image라는 변수에 받아줬었고, null이들어오면 noImg.png, NOTHING.JPG 써주기로 했음 따라서 삼항연산자를 통해 변수에 설정해주기

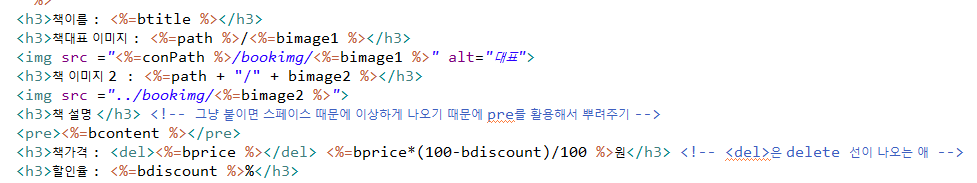
-- 나머지 애들은 mRequest.getPrameter("")로 받아주기

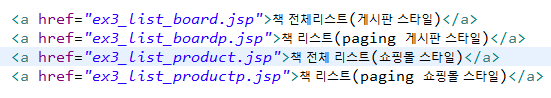


7-3)DB에 insert



--스크립트립 닫고, h태그들로 받아온 애들 화면에 뿌려보기





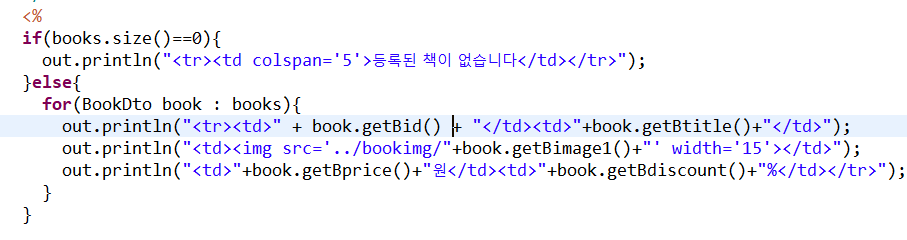
8. 책 전체리스트 (게시판 스타일)

**\*ex3\_list\_board.jsp**

-- dao 호출하고, listBook 불러오기

-- <table>생성해주고, 스크립트 열어서 books 사이즈가(listbook) 0이면 등록된 책이없는 것

사이즈가 0이 아니면 for문 돌려서 book.getBid()의 형태로 뿌려주기



9. 책 리스트 (paging 게시판 스타일 )

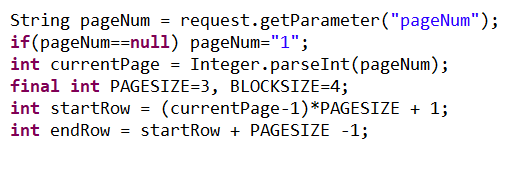
\* **ex3\_list\_boardp.jsp**

--pagesize=3으로, blocksize=4로 !\_!

--기본페이지는 ex3\_list\_board.jsp랑 똑같음. paging하는 공식만 잘 대입해놓기

-- pageNum 받아주고, starRow와 endRow 설정해준 뒤 listBook에 매개변수 있는 아이로

받아오기



--<table>밑에 페이지 넘어가는 숫자 들어가는 로직 만들어주기



10. 책 전체리스트 (쇼핑몰 스타일)

--우선 dao 불러오고, arrayList함수 불러오되 이번에는 확장 포문으로 뿌려주면 안됨 (밑으로 쭉 뿌릴게 니라서 )

-- 일반 for문 해서 3개씩 뿌릴거임

11. 책 리스트 (paging 쇼핑몰 스타일 )

--로직확인하기

12. 책상세보기 페이지

\*ex4\_detail.jsp

-- bid와 pageNum을 넘겨받아야 되니깐 getParameter 해주기

-- dao 부르고, getBook(bid)함수 가지고 오기