미니 프로젝트 과제물

1. 개요 및 요구사항

1-1. 개요

사용자로부터 URL로 구성되어있는 명령어를 입력받아 게시글을 작성할 수 있는 프로그램을 구성하려고 합니다. 게시글은 영구적으로 저장되지 않으며, 프로그램을 시작할 때마다 새롭게 저장하고 종료하면 게시글은 삭제됩니다.

1-2. 요구사항

요구사항의 단계별 문제를 해결하면 됩니다. 순서는 중요하지 않습니다. 해결할 수 있는 요구사항 먼저 해결한 다음 나머지 요구사항을 해결하면 됩니다. 높은 숫자의 단계에 해당하는 요구사항들은 그보다 낮은 숫자에 해당하는 요구사항에 대한 내용을 포함할 수 있습니다

만일, 특정 단계의 요구사항을 해결하기 다소 어려운 경우, 해결할 수 있는 범위에서 다른 내용을 해결하여도 좋습니다.

▼ 단계 1

1. 명령어를 입력받아서 출력할 수 있는 형태의 프로그램을 작성하세요.
    a. 명령어는 `명령어 >` 라는 문자 뒤에 공백 하나로서 받는 형태입니다.

    출력 예)

    ```
    명령어 > 안녕하세요!
    안녕하세요!
    명령어 >
    ```

2. 특정 명령어를 입력받았을 때 프로그램이 종료될 수 있도록 구성하세요.
    a. `종료` 혹은 `exit` 등 특정 명령어를 입력하였을 때에만 프로그램이 종료되도록 구성해 주세요.

    출력 예시)

```
명령어 > 안녕하세요!
안녕하세요!
명령어 > 종료
프로그램이 종료됩니다.
```

b. 종료되는 이외의 명령어를 입력하였을 때외에는 `존재하지 않는 명령어` 임을 사용자에게 알려주세요.

출력 예시)

```
명령어 > 안녕하세요!
존재하지 않는 명령어 입니다.
명령어 > 종료
프로그램이 종료됩니다.
```

3. 하기의 명령어에 맞추어 동작할 수 있도록 프로그램을 작성하세요.
   a. 게시글을 저장할 수 있는 객체를 정의해주세요.
   b. 다음은 작동하는 명령어와 그 동작에 대한 명세입니다. 정의한 객체를 아래의 명세에 맞추어 동작할 수 있도록 프로그램을 구성하세요.

명령어	**동작**	
작성	글의 제목과 내용을 입력받아 게시글을 작성할 수 있습니다.	
조회	마지막으로 작성한 게시글을 확인할 수 있습니다. 조회한 게시물은 다음과 같이 출력됩니다. 제목 : [작성시 입력한 제목] 내용 : [작성시 입력한 내용]	
삭제	마지막으로 작성한 게시글을 삭제합니다.	
수정	마지막으로 작성한 게시글을 수정합니다. 수정시, 제목과 내용이 모두 수정됩니다.	

4. 여러개의 게시글을 작성할 수 있도록 구성하세요.
   a. 각각의 게시글은 번호를 갖습니다. 번호는 작성할 때 마다 하나씩 증가하도록 구성하세요.

b. 게시글이 삭제되면 삭제된 자리는 다음 게시글이 위치합니다.

   i. 1번 2번 3번 게시글이 있을 때, 2번 글을 삭제하면 1번글 다음은 3번글이 되는 것입니다.

5. 게시물 기능을 다음의 요구사항을 만족하도록 수정하세요.

   a. 다음은 수정될 명령어와 그 명령어에 대한 동작 명세입니다. 3번에서 작성한 기능을 다음의 명세에 맞추어 수정해주세요

명령어	동작
조회	`조회` 명령어를 입력한 다음, 어떤 게시물을 조회할 지 번호를 추가적으로 받습니다. 다음과 같습니다. 명령어 > 조회 어떤 게시물을 조회할까요? 1번 1번 게시물 제목 : ... 내용 : ...
삭제	`삭제` 명령어를 입력한 다음, 어떤 게시물을 삭제할지 번호를 추가적으로 받습니다. 다음과 같습니다. 명령어 > 삭제 어떤 게시물을 삭제할까요? 1번 1번 게시물이 성공적으로 삭제되었습니다!
수정	`수정` 명령어를 입력한 다음, 어떤 게시물을 수정할 지 번호를 추가적으로 받습니다. 다음과 같습니다. 명령어 > 수정 어떤 게시물을 수정할까요? 1번 1번 게시물을 수정합니다. 제목: .. 내용: .. 1번 게시물이 성공적으로 수정되었습니다!

6. 게시물이 존재하지 않을 경우 예외가 발생하고 그 예외를 처리할 수 있도록 구성하세요.

a. 예외는 제공되는 예외를 사용하거나 직접 예외를 정의하여 사용하여도 무방합니다.

b. 존재하지 않는 번호의 게시글의 `조회`, `수정`, `삭제` 를 시도하면 해당 글은 존재하지 않는다는 취지의 예외가 발생합니다

c. 다음은 예외를 처리할 명령어와 그 명령어에 대한 동작 명세입니다.

명령어	동작
조회	존재하지 않는 게시글을 조회시, 다음과 같이 동작합니다. 명령어 > 조회 어떤 게시물을 조회할까요? 100번 100번 게시글은 존재하지 않습니다. 명령어>
삭제	존재하지 않는 게시글을 삭제시, 다음과 같이 동작합니다. 명령어 > 삭제 어떤 게시물을 삭제할까요? 100번 100번 게시글은 존재하지 않습니다. 명령어>
수정	존재하지 않는 게시글을 수정시, 다음과 같이 동작합니다. 명령어 > 수정 어떤 게시물을 수정할까요? 100번 100번 게시글은 존재하지 않습니다. 명령어>

7. 게시글 목록을 확인할 수 있는 기능을 구성해주세요.

   a. `목록` 을 입력하였을 경우, 게시글의 목록을 다음과 같이 출력해주세요.

```
명령어 > 목록
총 게시글은 3개 작성되어있습니다.

1번 게시글
제목 : ...
```

```
내용 : ...

2번 게시글
제목 : ...
내용 : ...

3번 게시글
제목 : ...
내용 : ...
```

▼ 단계 2

1. 명령어를 URL과 같이 구성하고, 프로그램의 구조를 변경하세요
    a. 명령어를 구분하는 `명령어 >` 부분을 `a` 로 변경해주세요
    b. URL을 분석할 수 있는 기능을 구현해주세요. 다음의 사항을 만족하도록 구현하면 됩니다.
        i. 입력받은 URL은 다음의 예)와 같이 입력되도록 구성합니다.

            예) `/구분/기능?파라미터...`

        ii. 파라미터는 실제 URL 규칙을 따릅니다. 단, 동일한 이름의 파라미터는 하나만 취급합니다. 가령, a라는 이름의 파라미터가 2개 이상 있을 경우 제일 마지막의 값만 저장합니다.
        iii. 유효하지 않은 URL이 입력되면 예외를 발생시킵니다. 발생시킬 예외는 자유입니다.
2. 게시판 기능을 구현하세요.
    a. 게시판 기능의 구분은 `boards` 입니다.
    b. URL을 기반으로 동작할 수 있도록 구현하세요.
    c. 다음은 게시판 기능의 URL과 그 동작입니다.

URL	파라미터	동작
/boards/add	-	새로운 게시판을 작성합니다.
/boards/edit	boardId	`boardId` 게시판을 수정합니다.
/boards/remove	boardId	`boardId` 게시판을 삭제합니다.

URL	파라미터	동작
/boards/view	boardName	`boardName` 게시판의 게시글 목록을 확인합니다. 다음과 같이 출력됩니다. 글 번호 / 글 제목 / 작성일 ... / ... / ... ... / ... / ...

d. 게시물 기능과 동일하게 예외발생 및 처리를 구성하세요

3. 게시물 기능을 수정하세요.
   a. 게시물 기능의 구분은 `posts` 입니다.
   b. URL을 기반으로 동작할 수 있도록 게시물 기능을 수정하세요
   c. 게시물 객체를 다음의 내용을 포함할 수 있도록 수정하세요
      i. 게시물 번호
      ii. 게시물 제목
      iii. 게시물 내용
      iv. 게시물이 속한 게시판
      v. 게시물 작성일
      vi. 게시물 수정일
   d. 다음은 게시물 기능의 URL과 그 동작입니다.

URL	파라미터	동작
/posts/add	boardId	`boardId` 게시판에 게시글을 작성합니다. 제목과 내용을 입력받으며, 작성된 시점이 저장됩니다.
/posts/remove	postId	`postId` 에 해당하는 게시글을 삭제합니다.
/posts/edit	postId	`postId` 에 해당하는 게시글을 수정합니다. 수정시 제목과 내용 모두 수정하며, 수정된 시점이 저장됩니다.

/posts/view	postId	`postId` 에 해당하는 게시글을 조회하며 다음과 같이 출력됩니다. [postId]번 게시글 작성일 : ... 수정일 : ... 제목 : ... 내용 : ...

4. 회원 기능을 구현하세요.
    a. 회원 기능의 구분은 `accounts` 입니다.
    b. URL을 기반으로 동작할 수 있도록 구현하세요.
    c. 해당 단계에서는 회원의 등급은 없습니다. 그러나 로그인을 했는지 하지 않았는지는 구분할 수 있어야 합니다.
    d. 다음은 회원 기능의 URL과 그 동작입니다.

URL	파라미터	동작
/accounts/signup	-	로그인을 위한 계정, 비밀번호, 그리고 이름 또는 닉네임, 이메일을 기반으로 회원을 등록합니다.
/accounts/signin	-	로그인을 위한 계정과 비밀번호를 기반으로 로그인을 수행합니다. 이미 로그인 되어있을 경우 예외가 발생하며, 로그아웃 하여야 새로운 계정에 로그인할 수 있습니다. 한 번에 하나의 계정만 로그인 할 수 있습니다.
/accounts/signout	-	현재 로그인 되어있는 계정을 로그아웃 처리합니다. 만일 로그인 되어있지 않은데 로그아웃을 시도하였다면 예외가 발생합니다.
/accounts/detail	accountId	`accountId` 에 해당하는 계정 정보를 조회하며 다음과

URL	파라미터	동작
		같이 출력됩니다. [accountId]번 회원 계정 : 이메일 : 가입일 :
/accounts/edit	accountId	`accountId` 에 해당하는 계정의 정보를 변경합니다. 비밀번호와 이메일만 변경할 수 있으며 변경일자가 기록됩니다.
/accounts/remove	accountId	`accountId` 에 해당하는 계정을 탈퇴(삭제)처리 합니다. 만일 로그인 되어있다면 로그아웃 처리를 먼저 수행합니다.

e. 로그인 후 게시물 혹은 게시판을 작성/생성하였을 경우 작성한 회원이 누구인지 저장되도록 수정하세요, 만일 로그인 되어있지 않은 사람이 글 혹은 게시판을 작성하였다면 비회원이라는 것을 표시하여야 합니다.

5. 다음의 요구사항에 맞추어 프로그램을 수정하세요
   a. 명령어를 입력받았을 때 요청(Request) 객체를 생성해주세요
   b. 요청 객체는 입력받은 명령어(URL)에 대한 정보를 포함하고 있습니다.
   c. 요청 객체는 로그인 여부를 포함하고 있습니다.
6. 다음의 요구사항에 맞추어 프로그램을 수정하세요
   a. 회원의 인증과 관련된 내용을 요청 객체와 분리하여 세션(Session) 객체에서 다룰 수 있도록 수정하세요
   b. 세션 객체는 요청 객체 안에 포함되도록 설계합니다
   c. 세션 객체 안에는 필요하다면 인증정보 외 다른 정보를 포함해도 무방합니다.

▼ 단계 3

1. 회원 기능에 등급을 추가하세요
   a. 회원은 관리자와 일반 회원으로 구분됩니다.
   b. 게시판은 관리자만 작성할 수 있습니다.
   c. 게시글을 로그인을 하여야만 작성할 수 있습니다.

d. 본인의 게시글만 수정하거나 삭제할 수 있습니다. 단, 관리자는 본인의 작성 유무와 관계 없이 게시글을 삭제하거나 수정할 수 있습니다.

e. 게시물은 로그인을 하지 않더라도 얼마든지 열람할 수 있습니다.

2. 컨테이너 객체를 도입하세요

   a. 컨테이너 객체에서 모든 객체를 생성하고 관리하도록 구성하세요.

   b. 모든 객체는 단 한 번만 생성되고, 그 생성된 객체를 사용하도록 설계하여야 합니다.

   c. 모든 객체에서 필요한 객체가 있다면 Container에 생성된 객체를 사용하도록 프로그램의 설계를 변경해주세요.

3. 필터(Filter) 기능을 추가하세요

   a. 입력받은 명령어(URL)을 필터링 할 수 있도록 구현하세요.

   b. 다음은 URL과 접근 권한에 대한 내용입니다. 다음과 같이 적용해주세요

URL	인증 여부	인가 범위
/accounts/signin	X	익명
/accounts/signup	X	익명
/boards/add	O	관리자
/boards/edit	O	관리자
/boards/remove	O	관리자
/posts/add	O	회원
/posts/edit	O	회원
/posts/remove	O	회원
/accounts/remove	O	회원
/accounts/edit	O	회원
/accounts/signout	O	회원
그외	-	-

▼ 추가 요구사항

1. 작성한 프로그램을 MVC(Model-View Controller) 패턴에 맞추어 수정하세요.

   a. 각각의 기능(게시판, 게시물, 회원)에 MVC 패턴을 모두 적용해주세요

   b. 각각의 기능이 분리되어 따로 실행될 수 있도록 결합도를 최대한 느슨하게 만들어 주세요