

1. 개요

이번 과제에서는 Linked List를 사용해서 영화의 장르와 제목이 저장되는 데이터베이스를 구현합니다. 이 데이터베이스에서는 삽입, 삭제, 검색 연산이 가능해야 하며, 각 항목이 영화의 장르와 제목에 따라 정렬된 순서로 저장되어야 합니다. Linked List를 이해하는 것이 과제의 목적입니다.

2. 뼈대 코드

뼈대 코드를 기본으로 하여 내용을 추가하도록 합니다.
이 코드에는 제출을 위한 입출력과 파일이름만이 정의되어 있습니다.

3. 데이터베이스 구조

우선 각 장르마다 영화 제목에 따라 정렬된 리스트를 만듭니다.
(ACTION, BATMAN BEGINS, ONG-BAK, THE MATRIX)
(DRAMA, MILLION DOLLAR BABY, THE AVIATOR)
(HORROR, HELLRAISER)
그리고 위의 리스트들을 원소로 가지며, 장르에 따라 정렬된 리스트를 만듭니다.
((ACTION, BATMAN BEGINS, ONG-BAK, THE MATRIX), (DRAMA, MILLION DOLLAR BABY, THE AVIATOR), (HORROR, HELLRAISER))

4. 지원하는 명령어(총 5가지)

0 삽입: INSERT %장르% %제목%

해당 장르와 제목을 가진 영화를 삽입합니다. 이미 데이터베이스에 있는 영화와 장르, 제목이 모두 같으면 삽입하지 않습니다.

1 삭제: DELETE %장르% %제목%

해당 장르와 제목을 가진 영화를 삭제합니다. 각 장르의 마지막 영화가 삭제되면 장르도 삭제됩니다.

2 검색: SEARCH %검색어%

제목에 검색어가 들어 있는 모든 영화들의 장르와 제목을 정렬된 순서로 출력합니다.

3 출력: PRINT

데이터베이스의 전체 내용을 정렬된 순서로 출력합니다.

4 종료: QUIT

프로그램을 종료합니다.

5. 입출력 형식

0 프로그램을 실행하면 한줄로 된 명령어를 입력받습니다.

1 명령어를 입력받으면 적절한 작업을 수행하고 SEARCH와 PRINT는 그 결과를 출력합니다.

2 명령어의 앞뒤 및 명령어와 인자들 사이에는 0개 이상의 공백이 들어갈 수 있습니다.(공백은 “ ”만 포함합니다.)

3 모든 영화들의 장르와 제목에는 '%'와 ',' 문자가 들어가지 않습니다.

4 영화 목록을 출력할 때에는 한 줄에 (장르, 제목)과 같이 하나의 영화만 출력하고, 결과가 더 있으면 다음 줄에 계속 해서 같은 형식으로 출력합니다. 데이터베이스가 비어있으면 EMPTY로 출력합니다.

5 SEARCH시에 검색어가 들어 있는 영화가 없다면, EMPTY로 출력합니다.

6 모든 입력은 대문자로 들어온다고 가정하셔도 좋습니다.

7 영화 목록을 정렬할 때에는 장르로 먼저 정렬하고, 장르가 같을 때에는 제목으로 정렬합니다. 참고로 데이터베이스를 제대로 구현한다면 목록을 정렬하기 위해서 별도의 작업은 필요치 않습니다.

8 출력할 때에는 한줄 한줄이 괄호로 묶여있어야 하며 ,(coma) 뒤의 공백 한칸을 반드시 띄워야 합니다.
아래 예시의 형식과 정확히 일치하게 만드세요.

9 한 명령어의 실행이 끝나면 QUIT를 입력받을때까지 다시 입력을 받습니다.

\$ java MovieDatabaseConsole	← 프로그램 실행
INSERT %ACTION% %BATMAN BEGINS%	← 이렇게 입력
INSERT %ACTION% %THE MATRIX%	← 이렇게 입력
INSERT %DRAMA% %MILLION DOLLAR BABY%	← 이렇게 입력
SEARCH %BA%	← 이렇게 입력 하면
(ACTION, BATMAN BEGINS)	← 이렇게 출력
(DRAMA, MILLION DOLLAR BABY)	← 이렇게 출력 한다.
DELETE %DRAMA% %MILLION DOLLAR BABY%	← 이렇게 입력
PRINT	← 이렇게 입력 하면
(ACTION, BATMAN BEGINS)	← 이렇게 출력
(ACTION, THE MATRIX)	← 이렇게 출력 한다.
QUIT	← 이렇게 입력 하면
\$	← 종료한다.

6. 유의사항

⑩ 장르와 영화 제목의 저장에는 Linked List만 사용하도록 합니다. Linked List는 직접 구현해야 합니다.

⑩ 프롬프트 (>)를 출력할 필요는 없습니다. 입력만 잘 받으면 됩니다. 프롬프트를 출력할 경우 출력결과가 이상해질 수도 있습니다.

⑩ 위의 명령어 형식을 잘 따르는 올바른 입력만 들어온다고 가정해도 좋습니다

니다. 참고로 명령어 형식을 따른 중복 삽입이나 데이터베이스에 없는 영화 삭제 및 검사 등은 올바른 입력에 해당합니다. 이러한 경우에는 위에서 설명한대로 적절한 조치를 취하시면 됩니다. (데이터베이스에 없는 영화 삭제나 중복 삽입은 따로 아무 메시지를 출력하지 않고 처리하면 됩니다.)

⑩ 주어진 스켈레톤 코드를 이용해 구현하여

\$ java MovieDatabaseConsole 같은 식으로 실행할 수 있도록 합니다. 클래스 내에서 함수를 더 추가하거나 private 클래스를 더 만드는 것은 상관 없습니다.

⑩ 제출 전, 반드시 컴파일 가능 유무를 확인합니다. 컴파일시 옵티마이제이션 옵션은 쓰지 않습니다.

⑩ 아래와 같은 명령어를 입력하면 컴파일이 이루어져야 하며, MovieDatabaseConsole 라는 이름의 클래스가 생성되어야 채점이 이루어집니다.

```
$ javac MovieDatabaseConsole.java
```

⑩ 컴파일 후, 일정 시간 초과시 프로그램을 강제 종료시키기 위해 다음과 같이 timeout 명령어를 반드시 사용합니다.

```
$ timeout [수행시간(초)] java MovieDatabaseConsole
    timeout 0.5 java MovieDatabaseConsole // 0.5초 수행
    timeout 1 java MovieDatabaseConsole // 1초 수행
```

⑩ 실행시간이 testset 기준으로 가상서버에서 200초를 초과할 시 감점이 있습니다. testset은 채점 시 새로 생성하므로 주어진 testset 실행시간과 다를 수 있습니다. 따라서 여유있게 완료하는 것을 권합니다.