

File Processing

Programming Project #3

1. Definition and Requirement for Implementation

본 프로젝트에서는 기존 프로젝트의 등록 리스트에 B-tree 자료구조를 사용하는 다단계 인덱스 (Multilevel Index)를 추가한다.

1.1 B-tree 인덱스 (70 Point)

1) 교재 9장과 부록 I의 BTree를 사용하여 구매 리스트에 인덱스를 생성한다.

- B-tree 인덱스를 통해서 데이터 파일(fileOfPurchase.dat) 레코드에 접근 한다.

- 관리자 모드만 고려함(일반유저 부분은 구현할 필요 없음)

- 수정이나 삭제, 삽입 시 인덱스 파일도 같이 변해야 함.

- 구매 리스트는 100개로 제한, B-tree의 차수(Order)는 4로 할 것.

- 인덱스 키는 문자열이 아닌 하나의 문자(char)를 사용할 것.

- 인덱스 파일의 이름은 fileOfPurchase.ind 로 한다.

- 부록 I의 BTreeNode, BTree 클래스를 사용하여 구현 할 것.

- 다단계(Multilevel), 기본 키(Primary key) 인덱스를 구현한다.

- 9장의 연습문제 13을 반드시 프로그램에 구현할 것.

- GamePurchaseSystem이 시작 될 때, 각각의 인덱스 파일의 존재 여부를 체크하여 인덱스 파일이 존재하지 않는다면 각각의 레코드 파일에 대한 인덱스를 생성하도록 한다. 초기에는 레코드 파일만 존재한다고 가정한다.

- 주의 : 모든 경우에 대해 참조 무결성이 유지되어야 함, 소스코드 주석 필수.

1.2 B+tree 인덱스 (20 Point)

- 1) 10장에서 소개된 Prefix B+ tree를 사용하여 **구매 리스트**의 인덱스를 생성한다.
- 2) 요구사항은 1.1 B-Tree인덱스와 동일하다.

2. Requirement for Document (10 Point)

- 1) 프로젝트 진행 중 자신이 필요하다고 생각되는 가정은 반드시 문서에 기술할 것
- 2) 구현한 클래스의 다이어그램(standard UML specification)을 그릴 것
- 3) 프로그램 실행 화면 캡처 및 동작 방법에 대한 설명 기술할 것.
- 4) 자신이 생성한 구매 리스트에 대한 간단한 설명
- 5) B-tree 인덱스를 생성하고 올바르게 동작하는 지에 대한 검증 과정을 화면 캡처와 소스코드 캡처, 결과화면, 파일 내용 캡처 등으로 기술할 것.
- 6) B-tree 인덱스를 통해서 데이터 파일 레코드에 접근하는 방식에 대한 설명
 - 소스코드 위치정보 포함(설명만 포함할 것, 소스코드 포함 X)

ex) main.cpp 파일의 20번째 라인부터 ~~함수 호출, 로직 등 이용하여 ~기능 구현함

- 7) B-tree 인덱스를 통한 삽입, 삭제 및 수정 방식에 대한 설명
- 8) 9장 13번 연습문제에 구현에 관한 설명
 - 코드 상에서 구현 위치와 작동 방식에 대한 설명
- 9) 프로젝트에서 사용된 소스 및 헤더 파일에 대한 설명
 - 제공되는 소스코드에 대한 설명은 1줄 이내, 자신이 작성한 파일에 대해서는 구체적으로 기술.

주의: 보고서에서 설명 및 동작에 대한 검증이 기술되지 않은 경우는 구현하지 않은 것으로 간주.

4. Due Date and Submission

1) 기한 : 2020. 12.16(수) 17:00 까지

2) 제출 방식

- E-mail : **fp2020fall@gmail.com**

제목과 파일 이름을 다음과 같은 형식에 맞추어서 제출

FP_PP_3_학번_이름 (예 : FP_PP_3_20201234_홍길동)

3) 제출 양식

- 첨부파일은 document와 프로젝트 파일을 같이 압축하여 제출한다. document는 압축 파일의 root에 있어야 한다.

- 제출파일은 반드시 zip 형식으로 압축한다. (예 : FP_PP_3_20201234_홍길동.zip)

- document는 한컴 한글이나 MS Word로 작성하며 **파일 제목은 메일의 제목과 같은 양식**을 지킨다. (예 : FP_PP_3_20201234_홍길동.doc)

- 프로젝트는 반드시 **MS Visual Studio 2017**에서 **Win32 콘솔 응용 프로그램**으로 진행하며, 제출할 때 **build clean**을 꼭 수행하고 **debug, ipch 폴더와 sql 파일을 삭제**한 후 압축하여 제출한다. **프로젝트의 이름도 메일 제목과 같은 양식**을 따른다. (예 : FP_PP_3_20201234_홍길동)

- 메일을 다시 제출하는 경우 메일 제목의 끝에 반드시 (재 제출) 이라고 명시한다.

- 제출 형식을 지키지 않는 경우 감점의 사유가 될 수 있다.

4. Assessment

- 1) 배점에 따라 기능 구현 여부, error handling, 대화형 방식의 적절성을 평가
- 2) 문서는 각 요구사항을 상, 중, 하로 구분하여 평가
- 3) 다음과 같은 경우 감점
 - 기한을 지키지 않은 경우. 당일 24시까지 제출하는 경우 20% 감점, 하루 late는 40% 감점, 이를 late는 70%감점하고 그 이후는 미 제출 처리
 - E-mail이나 hard copy 중 한 방식으로만 제출하면 50% 감점
 - 테스트 도중 segmentation fault 등의 비 정상적인 종료가 발생하는 경우 50% 감점
- 4) 다음과 같은 경우 0점
 - C++이 아닌 언어를 사용하거나, 객체 지향 프로그래밍을 사용하지 않은 경우
 - 첨부 파일에 바이러스가 있는 경우
 - 첨부 파일의 압축을 풀 수 없는 경우
- 5) 소스 코드 혹은 문서의 내용을 다른 사람과 Copy하는 경우 무조건 0점 처리하며, 이 후 두 번째 Copy가 적발되면 성적에 관계없이 F학점

5. Announcement

- 1) 질문 사항은 과목 게시판 (데이터베이스 연구실 홈페이지 게시판)을 이용할 것
- 2) 프로젝트 진행 중 발생한 모든 문제점에 대해 최대한 노력을 기울여 해결하고, 만약 해결할 수 없다면 문제점에 대해 논리적으로 문서에 상세히 기술할 것.