

File Processing

Programming Project #2

1. Definition and Requirement for Implementation (80 Point)

본 프로젝트에서는 기존 프로젝트에서 구현한 내용을 완성하고 관리자 모드 및 인덱스 기능을 추가하도록 한다.

1.1 관리자 모드 (20 Point)

1) 관리자 모드를 생성한다.

- GamePurchaseSystem의 첫 화면에서 회원 정보(Member)의 아이디(ID)와 패스워드(Password)로 로그인

- 관리자의 정보는 Member 객체로써 회원 정보 파일(fileOfMember.dat)에 저장

- Member 클래스에 level 필드를 추가. 관리자의 level은 1, 일반 사용자의 level은 9

- 관리자의 ID는 admin, Password는 adminpass로 기본 설정.

- 일반회원 테스트용 레코드를 첫 번째에 추가할 것.

(ID는 TestUser로 Password는 T1234로 할 것)

2) 관리자로 로그인 하는 경우 다음과 같은 기능을 사용할 수 있어야 한다. (관리자 메뉴 구성)

- 모든 회원(Member) 리스트에 대한 삽입, 수정, 삭제, 검색 기능

- 모든 게임(Game) 리스트에 대한 삽입, 수정, 삭제, 검색 기능

- 모든 구매(Purchase) 리스트에 대한 삽입, 수정, 삭제, 검색 기능

- 주의 : 모든 경우에 대해 참조 무결성이 유지되어야 함

3) 일반 회원으로 로그인 하는 경우 다음과 같은 기능을 사용할 수 있어야 한다. (일반회원 메뉴)

- 자신의 회원(Member) 정보에 대한 수정, 삭제(탈퇴) 기능
- 모든 게임(Game) 리스트에 대한 검색 기능
- 자신의 구매(Purchase) 리스트에 대한 검색, 삽입(등록), 수정, 삭제 기능
- 주의 : 다른 사용자와 관련된 모든 정보에 대해서는 검색, 수정, 삭제, 삽입 할 수 없다.
- 주의 : 모든 경우에 대해 참조 무결성이 유지되어야 함

1.2 인덱스 (60 Point)

1) 교재 7장과 부록 G를 사용하여 회원 리스트와 게임 리스트에 인덱스를 생성한다.

- 인덱스를 통해서 데이터 파일(fileOfMember.dat, fileOfGame.dat) 레코드에 접근 한다.
- 수정이나 삭제, 삽입 시 인덱스 파일도 같이 변해야 함.
- 인덱스 파일의 이름은 fileOfMember.ind, fileOfGame.ind로 한다.
- 부록 G의 **TextIndex, TextIndexBuffer, TextIndexedFile** 클래스를 사용하여 구현 할 것. 세 클래스를 모두 사용해야 하며, 필요한 부분이 있으면 추가로 구현해야 한다.
- 단일 단계, 기본 키 인덱스를 구현하며, 모든 인덱스가 메모리에 저장될 수 있다고 가정한다.
- 7장의 연습문제 13, 16, 17, 18을 반드시 프로그램에 구현할 것.
- **GamePurchaseSystem**이 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
- 주의 : 구매 리스트에 대한 인덱스는 다음 프로젝트에서 구현할 예정(B트리)

2. Requirement for Document (20 Point)

- 1) 프로젝트 진행 중 자신이 필요하다고 생각되는 가정은 반드시 문서에 기술할 것.
- 2) 구현한 클래스의 다이어그램(standard UML specification)을 그릴 것.
- 3) 각 프로그래밍 문제에서 요구한 사항을 빠짐없이 기술할 것.
- 4) 각 파일에 대한 기능 설명. 소스 코드는 포함하지 말 것.
- 5) 프로그램 흐름도 및 메뉴 이용 방법을 기술할 것.

3. Due Date and Submission

- 1) 기한 : 2020. 11. 25(수요일) 17:00까지

- 2) 제출 방식

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

제목과 파일 이름을 다음과 같은 형식에 맞추어서 제출

FP_PP_2_학번_이름 (예 : FP_PP_2_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_2_20201234_홍길동.zip)

- document는 한글이나 MS Word로 작성하며 **파일 제목은 메일의 제목과 같은 양식**을 지킨다. (예 : FP_PP_2_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 삭제한 후 압축하여 제출한다. **프로젝트의 이름도 메일 제목과 같은 양식**을 따른다. (예 : FP_PP_2_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, 이틀 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) 각 클래스는 앞으로 진행될 프로젝트에서 계속 사용되므로 반드시 재 사용성을 고려할 것
- 2) 질문 사항은 메일을 이용할 것
- 3) 프로젝트 진행 중 발생한 모든 문제점에 대해 최대한 노력을 기울여 해결하고, 만약 해결할 수 없다면 문제점에 대해 논리적으로 문서에 상세히 기술할 것.