**Design Project B**

Term Project #2

\* 블랙잭 프로그램을 작성하시오. 자세한 요구사항은 아래와 같다.

**1. System Requirements**

- 블랙잭 프로그램은 User, Card, Command, InitialCardDistribution, PersonalHitRequest, CheckLoser, CheckWinner, Replay, UserScoreInfo, CommandFactory, GameCommnadInvoker 그리고 App 클래스가 존재한다.

- User 클래스는 정수형 사용자의 아이디, 문자열 비밀번호, 문자열 이름, 정수형 자신의 남은 점수, 그리고 손에 들고 있는 카드(cardsInHand)들을 나타내는 Card 리스트를 가지고 있다.

- Card 클래스는 정수형 카드의 무늬, 정수형 카드의 번호를 가지고 있다.

- Command, InitialCardDistribution, PersonalHitRequest, CheckLoser, CheckWinner, Replay 클래스는 Command 패턴을 적용하기 위한 클래스로써, 사용자가 게임을 진행하면서 발생하는 이벤트들을 처리하는 클래스이다.

- Command 클래스는 문자열 커맨드의 이름과 순수가상함수 execute를 가지고 있다.

- InitialCardDistribution(플레이어에게 카드 2장씩 주는 명령)은 Command 클래스를 상속하여 구현한다. 모든 플레이어(로그인 한 User들을 의미)의 리스트와 남은 카드들의 리스트(큐로 구현)를 가지고 있다. Execute에 모든 플레이어들에게 카드를 2장씩 배포하는 기능을 구현한다.

- PersonalHitRequest(플레이어에게 카드 한장을 주는 명령)는 Command 클래스를 상속하여 구현한다. 모든 플레이어의 리스트와 남은 카드 리스트(큐로 구현), 그리고 요청한 플레이어의 인덱스를 가지고 있다. Execute에 카드 한 장 받기를 요청한 플레이어에게 카드를 한 장 배포하는 기능을 구현한다.

- CheckLoser(받은 카드들의 합이 21이 초과하는 플레이어를 게임에서 아웃시키는 명령)는 Command 클래스를 상속하여 구현한다. 모든 플레이어의 리스트와 등수를 계산하기 위한 우선순위 큐의 주소 값을 가지고 있다. Execute에 모든 플레이어의 리스트에서 패배한 플레이어를 확인하여 패배한 플레이어의 점수 정보를 우선순위 큐에 저장하고 패배한 플레이어를 플레이어 리스트에서 제거하는 기능을 구현한다.

- CheckWinner(게임 종료 후 플레이어들의 등수를 계산하는 명령)는 Command 클래스를 상속하여 구현한다. 모든 플레이어의 리스트와 등수를 계산하기 위한 우선순위 큐의 주소 값을 가지고있다. Execute에 플레이어 리스트에 남아있는 플레이어들의 점수 정보를 우선순위 큐에 저장하여 순위를 출력하고, 플레이어의 점수를 증가, 감소 시키는 기능을 구현한다.

- Replay는 Command 클래스(게임 플레이 중 발생한 모든 명령을 저장하고 재실행하는 명령)를 상속하여 구현한다. 하나의 게임 도중 생성된 모든 커맨드들을 저장하는 큐를 가지고 있다. Execute에 저장된 모든 Command들을 execute 하는 기능을 구현한다. 다른 커멘드들이 발생할 때마다 Replay 클래스의 커멘드 저장 큐에 enqueue를 한다.

- UserScoreInfo 클래스는 User 포인터 사용자 정보, 정수형 점수, 그리고 void calculateScore()를 가지고 있다.. calcuateScore 함수는 사용자 정보로부터 사용자가 가지고 있는 카드에 접근하여 점수를 계산하고 멤버 변수 점수에 저장한다.

- CommandFactory 클래스는 Command의 객체의 생성을 담당하는 클래스로써 아이디(커멘드의종류)를 입력 받아 아이디에 맞는 Concrete Command 객체를 생성하고 반환한다. CommandFactory 클래스는 메모리에 두개 이상 유지될 필요가 없는 클래스로써, 싱글톤 패턴을 적용하여 두 개 이상 유지되지 않도록 관리한다.

- App 클래스는 모든 사용자의 리스트, 로그인한 플레이어의 리스트, 리플레이를 위한 플레이어의리스트, 카드 리스트, 리플레이를 위한 카드 리스트, Invoker 객체, 등수 계산을 위한 우선순위 큐, 리플레이를 위한 우선 순위 큐를 가지고 있다.

App에서 게임 종료 후, 사용자가 배팅한 점수를 승자들에게 분배하는 기능을 하는 함수를 가지고 있다.

**2. 공통 기능**

**1) 로그인 기능**

- 로그인/로그아웃 기능은 게임을 플레이 할 사용자를 지정하는 기능을 한다. 사용자명과 비밀번호를 입력받아 로그인을 수행한다.

**2) 게임 플레이 기능**

**-** 게임 시작 전에 56장의 카드를 생성하고 셔플한다.

- 게임을 플레이 하는 기능을 구현한다. 게임 규칙은 다음과 같다.

1. **게임 시작 시 배당 점수를 설정**
2. **모든 플레이어에게 셔플 된 덱에서 로그인 순서대로 두 장씩 배포함**
3. **패배한 플레이어가 있는지 확인**
   * 패배 조건: 자신의 패의 카드 숫자의 합이 21 초과인 경우
   * 패배한 경우 점수를 0으로 세팅하여 등수를 저장하는 우선순위 큐에 저장
   * 패배한 플레이어를 플레이어 목록에서 제외
4. **플레이어 리스트 순서대로 플레이어 중에서 카드를 한 장 더 받을지 물어봄**
5. **패배한 플레이어가 있는지 확인**
6. **모든 플레이어가 카드를 그만 받을 때 까지 4~5번을 반복**
   * 모든 플레이어가 카드를 그만 받으면 모든 사용자의 점수를 계산하여 등수를 저장하는 우선순위 큐에 저장함

* **승리자를 제외한 모든 플레이어가 승리자에게 배당 점수를 지급**
  + 모든 플레이어가 패배한 경우 모든 플레이어의 보유 점수를 배당 점수 만큼 차감
  + 승리자가 다수인 경우, 패배자가 공여한 배당 점수를 승리자가 똑같이 나눠 가짐
  + 전원 승리자인 경우는 무승부로 점수의 차감 없음

- 2, 3, 4, 5 단계는 Command 패턴을 사용하여 처리한다.

**3) 리플레이 기능**

- 게임 종료 직후, 이전 게임을 다시보기 하는 기능을 구현한다.

- Replay 클래스를 사용하여 이전 게임에 발생한 모든 Command의 목록을 통해 이전 게임을 다시보기 할 수 있다.

**3. 파일 입출력**

- user 파일: 각 사용자의 정보를 입력 및 출력 파일

UserUser\_id password username remainScore  
1 pw1 홍길동 10000  
2 pw2 김철수 9800

**4. 프로그램 요구사항**

- 커맨드 패턴: 플레이어의 요청을 커맨드 패턴을 적용하여 처리한다.

- 팩토리 패턴: Command 객체를 생성하여 반환하는 클래스를 구현한다.

- 싱글톤 패턴: 팩토리 클래스를 메모리에 두개 이상 유지하지 않도록 구현한다.

- 프로그램 종료 시 변경사항을 파일에 적용하여야 한다.

**게임 재생 흐름**

1. start 명령어를 입력하여 게임을 시작합니다.
2. 각 플레이어에게 초기 카드를 분배합니다.
3. 플레이어가 히트를 요청할 수 있습니다.
4. 패배자를 체크하고 패배한 플레이어를 제거합니다.
5. 모든 플레이어가 카드를 받지 않으면 승자를 체크합니다.
6. 게임이 끝나면 Replay 명령어를 통해 이전 게임을 재생할 수 있습니다.