**Design Project A**

Project #2

\* 우편물 배송 관리 프로그램을 작성하시오. Client는 자신의 우편물을 다른 Client에게 보낼 수 있고, 또한 다른 Client가 보낸 우편물을 받을 수 있다. 우편물 배송 시스템은 전국적으로 여러 개의 우편집중국으로 구성된다. 각 우편집중국은 이웃의 우편집중국들만 연결되어 있어, 멀리 있는 우편집중국은 여러 개의 우편집중국을 거쳐서 보내야 한다. Manager는 이와 같은 우편집중국 시스템을 관리한다. 자세한 요구사항은 아래와 같다.

**1. System Requirements**

- 우편물 배송 관리 프로그램을 구현하기 위한 클래스는 App, User, MailCenter, Mail 이다.

- User 클래스를 상속받은 사용자 클래스 Client와 우편물 관리자 클래스 Manager가 있다.

- Mail 클래스를 상속받은 편지 클래스 Letter와 소포 클래스 Package가 있다.

- App은 User DB, Router DB, Mail Center DB를 가지고 있다.

- User는 아이디, 비밀번호, 사용자 타입을 가지고 있다.

- User를 상속받은 Client는 이름, 주소, 보낸 우편물 목록, 받은 우편물 목록을 가지고 있다.

- User를 상속받은 Manager는 추가적인 맴버변수는 가지고 있지 않다.

- Client와 Manager는 User클래스의 순수가상함수 send( )와 receive( ) 함수를 overridding하고 있어 서로 다른 기능을 수행한다.

- App는 User와 MailCenter 객체들을 총괄하여 관리하고 필요한 함수를 호출한다.

- 프로그램 실행 시에 User DB, Router DB, Mail Center ID DB를 App 클래스의 각 컨테이너에 로드한다.

- 프로그램 종료 시에 갱신된 정보들을 파일에 다시 쓰지 않는다. 이번 프로젝트는 난이도를 고려하여 모든 우편물 송수신을 메모리에서만 동작하는 것으로 고려하고, 파일 입출력을 하지 않는다. 단, 앞서 언급한 우편물 관리에 필요한 모든 DB들은 프로그램 시작 시 읽어와야 한다.

- 메인에서는 오직 App의 run()만을 호출한다.

**2. 공통 기능**

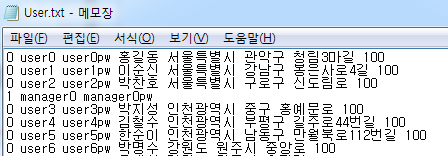
* 데이터 적재 기능 : 프로그램 시작 시, User.txt, Router.txt, MailCenterID.txt로부터 데이터를 읽어와 user\_list, router, mcenter\_ID\_list에 각각 로드한다.
  + User DB

User.txt 파일에는 User를 상속받은 Client와 Manager가 섞여 저장되어 있다. 따라서 첫 번째 필드인 userType(0==Client, 1==Manager)을 읽고, 해당하는 클래스의 객체로 읽어와야 한다.

Client와 Manager는 공통적으로 부모인 User가 가지고 있는 userType, ID, pw를 가지고 있다.

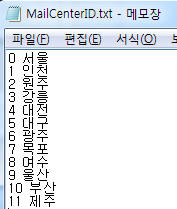
Client는 name, address를 추가적으로 가지고 있다. (앞서 언급한 것과 같이 우편물 송수신은 메모리기반으로만 고려하므로 sentMailList, receivedMailList는 파일에 입출력을 하지 않는다.)

Manager는 한 명만 존재하며, User 클래스의 멤버변수 이외의 추가 필드를 가지고 있지 않다.



* + Mail Center ID DB

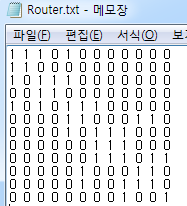
MailCenterID.txt 파일에는 전국의 우편집중국과 각 우편집중국의 ID 필드가 있다. 이 파일을 읽어 mcenter\_ID\_list에 로드한다. ID 정보는 다음에 소개할 Router.txt 파일의 이차원 배열의 인덱스 번호와 일치한다. 우편집중국 이름(서울, 인천, 광주 등)은 MailCenter 객체를 생성할 때 사용한다. 즉 이 파일의 정보는 우편집중국의 ID와 이름을 매핑하는 역할로 다양하게 활용될 수 있다.



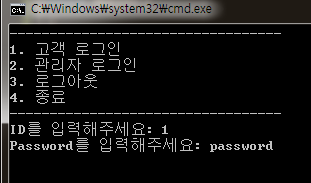
* + Router DB
    - Router.txt는 우편집중국들 간에 연결정보를 가지고 있다. 이 파일의 정보를 읽어 이차원 배열인 router를 채운다. 아래의 그림과 같이 각 우편집중국은 일부의 이웃들과만 연결되어 있다. 이와 같은 연결 정보는 그래프로 모델링이 가능하며, Router파일은 인접행렬(adjacency matrix)로 연결정보를 표현한 자료구조이다.



* + - 아래는 우편집중국간에 연결 정보를 나타내는 이차원 행렬이다. 예를 들어 0번 행이 2번 행과 연결되어 있다(2차원 배열 (0,2)==1). 이 의미는 앞서 설명한 MailCenterID.txt 파일에서 0번은 서울을, 2번은 원주를 의미하므로 서울과 원주는 우편집중국이 연결되어 있다는 것이다.
    - 따라서 임의의 지역에서 다른 지역으로 우편물을 보낼 때는 한번에 보낼 수 없는 경우가 발생한다. 이 경우 중간 우편집중국들을 경유해서(routing) 보내야 한다. 이때 다음 우편집중국을 선택하는 알고리즘인 최단경로탐색(shortest path discovery)이 필요하며, 이 코드는 함수형태로 제공할 것이다.

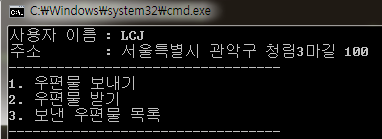


* 로그인 : 프로그램은 사용자로부터 아이디와 비밀번호를 입력 받는다. 전체 User의 정보와 입력 값을 비교하여 일치하는 정보를 찾는다. 일치하는 경우 해당 정보를 현재 로그인한 사용자로 가지고 있는다. 현재 로그인한 사용자는 User\* 형태의 변수로 선언된다.

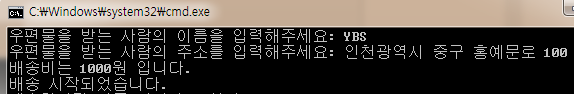


**[고객 기능]**

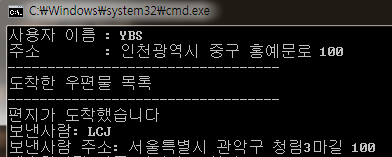
* 다음 아래의 기능들을 로그인한 고객에게 제공한다.



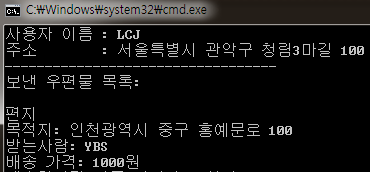
* 우편물 보내기: 현재 로그인 된 사용자가 우편물을 보낸다.



* 우편물 받기: 현재 로그인 된 사용자에게 온 우편물을 받는다.

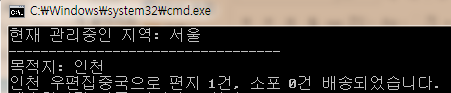


* 보낸 우편물 목록:

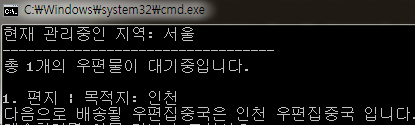


**[관리자 기능]**

* 다음 아래의 기능들을 로그인한 관리자에게 제공한다.
* 우편물 보내기: 현재 관리중인 우편집중국에 저장된 우편물들을 고객 또는 인근  
  우편집중국으로 발송한다.



* 우편물 받기: 분류 대기중인 우편물들을 분류한다.



**[테스트]**

* 최종 테스트 시에 사용자 및 우편집중국은 주어진 DB파일로 테스트한다.

**[특수 상황 조건]**

* 실제 프로그램과 약간의 차이로 발생하는 특수한 상황의 결정은 본인이 직접 규칙을 정하고, 발표 시 설명한다.

**3. 예외 처리**

- 자기 자신에게 우편물을 발송 할 수 없다.

- 우편물을 받는 고객의 이름이 존재하지 않거나, 우편물의 목적지 주소가 존재하지 않거나,  
 받는 고객의 주소와 우편물의 목적지 주소가 일치하지 않는 경우엔 발송할 수 없다.

- 메뉴 선택 시 메뉴의 리스트 이외의 값을 넣었을 경우

- 잘못된 아이디 혹은 비밀번호로 로그인 시도 시

**4. Program Requirements**

- App , User, Client, Manager, Mail, Letter, Package, MailCenter 클래스로 구현

- 소스코드에 함수, 중요 변수, 중요 기능, 중요 알고리즘 등은 반드시 주석을 단다.

**5. Interface Example**

- 자신이 편리한 인터페이스를 자유롭게 디자인한다.