김태경

객체지향 시스템 설계

2022년 4월 26일

**Assignment1**

1. **실행 방법**

WSL과 jdk 17환경에서, ‘assignment1’패키지가 있는 경로로 이동 후

javac ./assignment1/PatientManager.java

명령어 통해 클래스를 생성한 후,

java assignment1.PatientManager

명령어 통해 코드 실행할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

1. **구조 설계 및 methods**
   1. **클래스**(Patient, PatientManager, User, UserManager)

**Patient:**

환자의 ID와 이름, 나이, 상태 정보를 저장하는 클래스이다.

환자의 정보가 담긴 변수들은 private 접근제어자를 통해 다른 클래스에서 접근하지 못하도록 하여 보안성을 높였다.

생성자를 통해 환자의 정보를 초기화 할 수 있다.

**PatientManager**:

Main method가 동작하는 클래스이다.

PatientList를 정적변수를 통해 선언하여, Patient 클래스를 저장해 인스턴스를 생성하지 않아도 호출될 수 있도록 한다.

ind와 scan도 같은 static으로 선언됐다.

**User:**

로그인을 위한 아이디와 비밀번호를 저장하는 클래스이다.

Patient 클래스와 같이 클래스의 변수들은 private 접근제어가 돼있으며, 생성자를 통해 초기화가 가능하다.

**UserManager:**

PatientManager 클래스와 같이 User 클래스들을 저장하는

UserList와 scan을 정적으로 선언했다.

* 1. **함수**

**Patient:**

**getPid-** 환자의 아이디를 리턴한다.

**setName, setAge, setSatatus-** 함수를 통해 환자의 이름과 나이, 상태 정보를 재설정 할 수 있다.

**toString-** 함수를 통해 환자의 정보들을 문자열로 변환하여 리턴한다.

**PatientManager**: (static 변수를 사용하는 모든 함수에 static이 정의된다)

**managePatients**- 메인 페이지를 보여주는 함수로, scanner를 통해 입력 받은 값으로 옵션을 선택하여 다른 함수를 할 수 있으며, 다른 함수의 실행이 종료되고 나서while문을 통해 다시 선택화면으로 돌아가거나 종료가 가능하다.

**addPatient-** scan을 통해 입력 받은 환자의 정보를 Patient 클래스를 통해 생성하여 바로 patientList에 저장한다.

아이디를 입력하고 난 후 doesExist함수를 통해 아이디가 이미 존재하는지 확인을 하고, 아이디가 존재하지 않을 경우 계속 정보 입력이 가능하며, 아이디가 존재하는 경우 addPatient함수는 바로 종료된다.

환자를 저장할 때마다 patientList의 인덱스인 ind가 1씩 커진다.

**doesExists-** 파라미터로 환자의 아이디 값을 받은 후 patientList에 해당 아이디가 존재하는지 확인하고, 존재하는 경우 true값을 리턴하고, 존재하지 않는 경우 false를 리턴한다.

for문에서는 patientList 값이 null이 아닐 때에만 내용을 확인한다.

**outputAllPatients-** patientList의 환자 정보를 출력한다.

for문을 통해 환자의 정보 존재여부를 확인 후 환자가 있을 때 분류내용을 출력한 후, exist를 true로 바꾸고once를 true에서 false로 바꿔 한번 만 출력될 수 있도록 하고 환자의 정보를 출력한다.

for문이 끝난 후 환자정보가 없어 exist가 계속 false일 경우 환자의 정보가 없다는 내용을 출력한다.

**deletePatient-** 환자의 정보를 삭제하기 위해 삭제할 아이디 값을 입력 받은 후 delete 함수를 호출한다. delete함수를 통해 삭제가 되지 않았을 경우 해당 아이디가 존재하지 않는다는 내용을 출력한다.

**delete-** 파라미터로 환자의 아이디 값을 받은 후 for문을 통해 동일한 아이디를 찾을 경우 해당 환자의 null로 변경(정보 삭제) 후 true값을 리턴하고, 동일한 아이디가 없을 경우 false값을 리턴한다.

**updatePatient-** 환자의 아이디 값을 입력 받은 후 doesExist함수를 통해 아이디 존재여부를 확인하고, 아이디가 존재할 경우 set함수를 통해 환자의 이름과 나이, 상태를 수정한다.

아이디가 존재하지 않을 경우 해당 아이디가 존재하지 않는다는 내용을 출력한다.

**User:**

**getUid-** uid를 리턴한다.

**getPassword-** 비밀번호를 리턴한다.

**UserManager:**

**addUser-** userList에 User클래스 생성자를 통해서 값을 저장한다.

**login-** 아이디와 비밀번호를 입력 받은 후 for문을 통해 userList에서 해당 아이디가 존재하고 비밀번호가 일치하는지 확인한다.

로그인이 제대로 됐을 경우 true를 리턴하며, 로그인에 실패했을 경우 false를 리턴한다.

* 1. **Main**

맨 처음 UserManager를 선언한 후 addUser함수를 통해 계정을 추가한다.

그리고 login함수를 실행하여 로그인을 한 후 리턴받은 true값으로 managePatients함수를 실한다.

로그인에 실패할 경우 바로 프로그램은 종료된다.