20191611 유종선 과제 7 확장된 성적처리시스템의 구현

* 코드

1. Student.py

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

1. GradingSystem.py

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* 결과

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

* 먼저 student.py에서는 조건으로 명시한대로, Student 클래스를 만들어서 Person 클래스를 상속받도록 하였다. 각 속성은 보호모드로서 @property를 이용하였다.
* Main에서 5명의 학생 정보를 입력받아서, Student클래스 학생 객체를 생성하였다.
* 이후에, GradingSystem 클래스 내의 register\_student 메서드로 등록시켰다.
* Student.py와 GradingSystem.py는 각각 다른 파일에 작성되었다.
* Register\_student 메서드에서 학생 정보를 등록하였다. 딕셔너리를 사용하여 학생의 정보를 담아두었다.
* Process 메서드는 학생들의 총점에 따라 등수를 계산하는 메서드이다. 조건에 따라 lamda 함수를 사용하여 정렬하였다. 또, 추가적인 기능을 위해 각 과목별로도 정렬을 해주었다.
* Print\_student메서드는 전체 학생 명단을 프린트하는 메서드이다. 이때, 등수에 따라 정렬되어 print된다.
* Print\_class\_info는 반 정보를 프린트하는 메서드이다. 과목별로 총점을 계산하고 평균을 계산하였고, 각 과목별 최고점을 도출하고, 최고점을 맞은 학생의 정보를 print하였다.
* 여기서부터는 부가기능이다.
* findStudent 메서드는 학생을 찾아주는 메서드이다. 학생의 이름을 입력하면, 그 이름을 가진 학생의 정보를 출력한다. 찾았다면, 주민번호를 토대로 그 학생의 나이와 성별을 계산하여 출력한다. 만약 찾지 못했다면, 찾지 못했음을 출력한다.
* Rank 메서드는 과목의 rank를 보여주는 메서드이다. 사용자가 보고 싶은 과목을 입력하면, 그 과목에서의 학생들의 등수를 계산한다. 역시 같은 점수일 경우에는 같은 등수로 처리해주었다.
* 마지막으로 전체 실행시간을 decorator를 사용하여 구현하였다.