텍스트이(가) 표시된 사진

자동 생성된 설명

위의 사진은 Student 클래스를 StudentFile.py 라는 파일에 저장하는 코드이다. %%writefile 이라는 magic command를 사용하여 파일을 생성하였다. Student 클래스는 인자로 번호, 이름, 국어점수, 영어점수, 수학점수를 입력받아서, 인스턴스 변수로서 저장하고 있는다. 이 과저은 \_\_init\_\_ 함수 안에서 일어나는데 밑줄 하나짜리 인스턴스변수로 설정하여 보호모드로 사용할 수 있도록 하였다. 보호모드로 설정된 인스턴스 변수는 매서드로서만 접근하도록 암묵적으로 이야기해준다. 따라서, 보호모드 인스턴스 변수에 접근하기 위해 접근을 위한 매서드들을 정의하였다. 그리고 이 매서드들을 실제 변수에 접근하는것러럼 사용하기 위해 @property를 사용하였다. 모든 인스턴스 변수는 모두 get 매서드를 정의하였고, 등수만 set 매서드를 정의하였다. StudentGradingSystem 클래스의 process 매서드에서 계산하여 등수를 새롭게 지정해주어야 하기 때문이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

위의 사진들은 StudentGradingSystem 클래스를 StudentGradingSystem.py 라는 파일에 저장하는 코드이다. 이 클래스에서는 학생 객체 등록, 등수 계산, 학생정보, 반 정보 출력 매서드가 정의되어 있다. 먼저, 학생 객체 등록이다. 앞서 학생 객체는 Student 클래스를 이용하여 생성하였다. 그렇게 생성된 학생 객체를 인자로 받아서 StudentGradingSystem 클래스의 클래스 변수인 stuObjs에 추가한다. 모든 학생이 등록이 완료가 되었다는 의미는 모든 학생정보가 stuObjs 리스트에 추가되었다는 것이다. 이렇게 등록이 완료가 되었다면, process 매서드를 이용하여 등수와 반 전체 정보를 얻어온다. 등수를 구할때는 생성된 stuObjs 리스트를 총점 순으로 내림차순 정리를 가장 먼저 진행하였다. 이렇게 정렬을 할때 lambda 함수를 활용하였다. 즉, sort함수의 key에서 lambda 함수르 활용하였다. 기본적으로 sort는 오름차순이므로 lambda 함수 내부에서는 점수에 -를 붙여주어 내림차순 정렬이 될 수 있도록 프로그램을 작성하였다. 정렬이 되었다면 첫번째 학생이 가장 시험점수가 높고, 마지막 학생이 가장 시험 점수가 낮다. 이렇게 만들고 반복문으로 한명씩 불러와서 등수를 1씩 올려가며 할당해주고싶지만 동점자를 고려해야하기 때문에 이렇게 할 수는 없다. 현재 등수 변수와 동점자 변수를 둘다 두고, 동점자가 등장했다면 동점자 변수를 1 증가시키고, 더 낮은 점수를 가진 학생이 등장했다면 현재 등수변수에 동점자 변수에 저장된 수를 더한 후, 그 등수를 학생에게 할당해주고, 동점자 변수는 다시 1로 만들어준다. 그리고 매번 그러한 과정이 끝나면 현재 점수를 직전 점수로 바꿔주고 그 다음 학생의 점수를 분석하러 간다. 첫번째 학생은 항상 1등이고, 그 이후의 학생들에 대해서 위의 과정을 반복해준다. 다음으로는, 반 전체 성적 정보를 구하러 가는 과정이다. For loop 으로 학생 객체를 하나씩 살펴보며 국어, 영어, 수학의 최댓값을 찾아내고, 그러한 최댓값을 가지는 학생들의 명단을 리스트에 저장하였다. 동시에, 총점, 국어, 영어, 수학 점수를 클래스변수로 정의된 변수에 매번 더하여, 총점, 국어점수, 영어점수, 수학점수의 합을 구한다. 그 이후 모든 합들을 학생 수로 나누어 평균을 구한다. 다음으로 print\_students 매서드에서는 학생정보를 출력하고, print\_class\_information 매서드에서는 반 전체의 성적 정보를 출력한다.

텍스트이(가) 표시된 사진

자동 생성된 설명



먼저, Student 클래스와 StudentGradingSystem 클래스는 각각 StudentFile.py 와 StudentGradingSystem.py 라는 다른 파일에 저장되어 있다. 이 클래스들을 활용하기 위해서는 파일을 import 시켜와야한다. 그리고 time 모듈을 import 하여 실행시간을 측정하고자 하였다. checkTime 함수는 OOPstudentGrading 함수의 실행시간을 측정하기 위한 데코레이터이다. 시작시간과 종료시간을 얻은 후, 그 차이를 구하여 실행시간을 측정하였다. 그리고 OOPstudentGrading 함수는 주어진 텍스트 파일을 열어서 그 안의 학생 정보를 위에서 만들어놓은 클래스들을 활용하여 처리하는 함수이다. 가장 먼저, SData.txt 라는 파일을 텍스트 읽기모드로 연 후, for loop을 이용하여 각 줄에 접근하여 문자열 형식으로 받아온 줄의 내용을 공백을 구분자로 하여 각각의 원소들을 data 리스트에 저장한다. 그렇게 하게 되면 번호, 이름, 국어점수, 영어점수, 수학점수를 추출해내기 용이해진다. 그렇게 얻어왔다면, Student 클래스를 이용하여 학생 객체를 생성하고, StudentGradingSystem 클래스의 register\_student 매서드를 통해 등록한다. 등록이 완료되었으니, StudentGradingSystem 클래스의 process 매서드를 호출해서 학생들의 등수를 구하고, 반 전체 성적에 대한 정보를 구한다. 그리고 print\_students 매서드로 학생정보를 출력하고, print\_class\_information 매서드로 반 전체의 성적 정보를 출력한다.