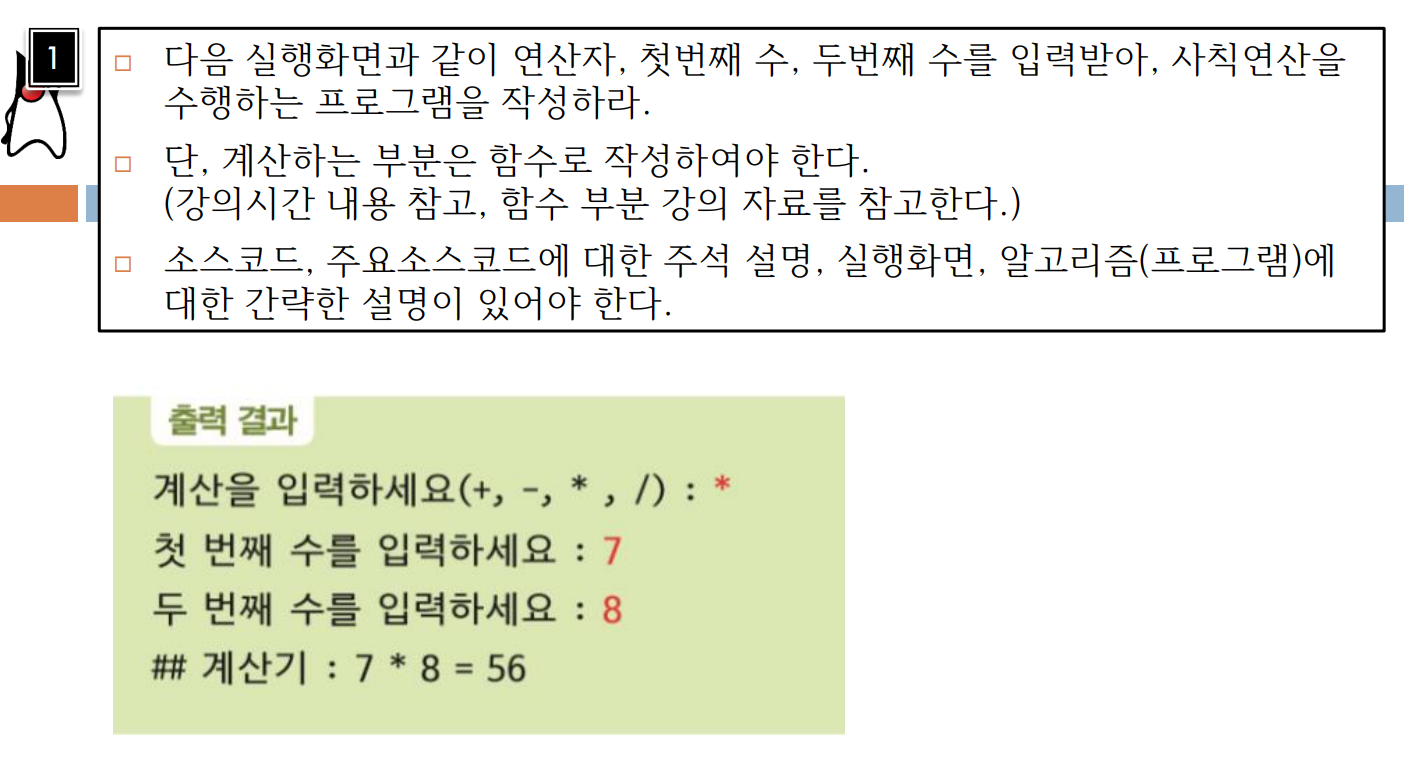
1문제)



소스코드)

텍스트이(가) 표시된 사진

자동 생성된 설명

실행화면)

텍스트이(가) 표시된 사진

자동 생성된 설명

풀이)

두 수와 계산할 부호를 넘겨받아 계산해주는 calc라는 함수를 정의한다. 만약 부호가 “+”이면 두 수를 더한 값을 반환하고 나머지 “-“, “\*”, “/”도 동일하게 비교하여 반환한다. 그리고 이 네 부호를 입력하지 않고 다른 입력을 할 가능성이 있기 때문에 예외처리로 else문을 사용해서 “다시 입력하세요”를 출력하고 0 값을 리턴하였다. 메인 코드에서는 먼저 함수를 호출할 때 매개변수로 넘길 부호와 두 수를 입력 받는다. 그 뒤에 result에 calc 함수에서 얻는 리턴 값을 저장한다. 이렇게 얻은 결과 값들을 출력해주면 프로그램을 끝이 난다. 복잡한 코드를 짜기 시작하면 함수들을 이용하는 것은 필수가 되고 여러 번 쓰이는 코드들을 객체화 시켜 메인 절에서 호출하는 경우가 많기 때문에 가시성 있게 함수를 짜는 것이 중요하다고 생각한다. 그리고 나중에 클래스 안에 여러가지 함수(메서드)들이 존재하고 함수를 상속해서 입맛에 맞게 약간 변화해서 쓰는 경우도 있기 때문에 항상 변수 이름을 의미 있게 쓰려고 노력하고 코드 들여쓰기를 예쁘게 하고 주석을 써서 보다 자세히 설명해 가시성을 높이려고 노력하고 있다.

2문제)

텍스트이(가) 표시된 사진

자동 생성된 설명

소스코드)

텍스트이(가) 표시된 사진

자동 생성된 설명

실행화면)

텍스트이(가) 표시된 사진

자동 생성된 설명

풀이)

점수를 넘겨받아 등급을 계산해주는 함수 getGrade를 만들어준다. if문을 이용하여 90점 이상이면 “A”를 반환하고 나머지 “B”, “C”, “D”도 동일하게 비교하여 반환한다. 이 외의 수를 입력하면 else문을 이용하여 “F”를 반환한다. 100점 이상 입력하거나 음수를 입력할 경우 예외처리를 할 수 있지만 보통 그럴 경우는 없을 것 같아 별도로 예외처리를 하지는 않았다. 메인 절에서는 먼저 매개변수로 넘길 score를 실수형 float으로 형 변형하여 입력 받았다. 점수는 실수일 경우도 많기 때문이다. getGrade에서 얻은 리턴 값을 grade 변수에 저장하고 이 값을 출력해주었다. 역시 함수는 정말 매력적인 것 같다. 자동차 조립할 때 메인 뼈대에서 핸들 바퀴 등등을 모듈화해서 조립하는 것처럼 프로그램도 객체화해서 함수로 메인 절에서 여러 함수들을 호출해서 사용할 수 있어 재밌는 것 같다. 아직 여려 명에서 개발해본 경험은 없어 아쉽지만 앞으로 여려 명에서 개발해보고 싶다.