

## 객체지향 프로그래밍 소스 구현 설명

202104361

ICT공학부

천승환

1. 문제 정의 : 10명의 학생들의 성적을 scores배열에 저장하여 60점 이상으로 통과한 학생의 수를 리턴받아 출력한다.

2. 문제 해결 방법 :

(1)번 문제. Dept com(10)을 통해 학생 수를 입력받아 scores 배열을 동적 할당. 복사 생성자를 이용해 size와 scores 배열을 복사. 소멸자로 동적으로 할당된 scores배열의 메모리 삭제. countPass 함수는 복사 생성자를 이용하여 dept 객체가 복사되도록 함. 마지막으로 isOver60함수를 통해 60점 이상이면 true를 반환, 그게 아니면 false로 반환.

(3)번 문제. 복사 생성자를 제거하여 최소한의 코드로 다시 짜는 것이므로 간단하게 int countPass(Dept dept)에서 (Dept& dept)로 바꿔 사용하면 된다.(참조에 의한 호출 사용)

3. 아이디어 평가 :

(1)번 문제. 복사 생성자에서 기존 객체인 size와 scores 배열을 복사하는데, 깊은 복사를 통해 scores배열을 새로 할당하고 데이터를 복사하도록 하였다. 그리고 논리 자료형인 bool함수를 사용함으로써 참과 거짓을 리턴하여 메인 함수의 변수에 할당하도록 하였다.

(3)번 문제. countPass함수가 실행될 때, 기존에 있던 객체가 아닌 복사된 객체를 통해 코드가 돌아가기 때문에 불필요한 메모리가 할당된다. 따라서, 복사 생성자를 사용하여 전달하는 것 보다 countPass함수의 매개 변수를 참조로 전달하게 되면 메모리를 낭비하지 않아도 된다.

4. 문제를 해결한 키 아이디어 또는 알고리즘 설명 :

(1)번 문제. 복사 생성자를 사용하는 것이 키 아이디어라고 생각한다. 기본적으로는 객체를 복사할 때 "얕은 복사"가 이루어져, 배열의 주소만 복사되는데, 이 문제의 코드는 "깊은 복사"이기 때문에 새롭게 메모리를 할당하고 데이터를 복사하여, 각 객체들이 독립적으로 동작하도록 했다는 것이다.

(3)번 문제. 복사 생성자를 제거하여 프로그램 코드를 짜는 것인데, 이에 대한 해답은 참조를 통해 객체를 전달하는 것이다. 기존에 있던 객체에 직접적으로 접근하여 사용하면 메모리 성능을 높일 수 있다.