

문제 정의 :

이미 구현되어 있는 부분에 더해서

결과가 다음과 같이 되도록 클래스 멤버를 구현하고 프로그램을 완성하자.

10 개의 점수 입력>> 10 20 30 40 50 60 70 80 90 100

60 점 이상은 4 명

Dept 클래스에 복사 생성자를 제거하자. 복사 생성자를 제거해도 오류가 발생하지 않도록 프로그램을 수정하자.

문제 해결 방법 :

1 번

구현되어 있지 않는 부분은 Dept 복사생성자와 Dept 소멸자, read 함수와, isOver60 함수의 구현부이다.

먼저 Dept 복사생성자는 countPass 함수에 매개변수로 Dept 가 들어가는 부분에 사용된다고 생각했다. Scores 변수가 포인터로 되어있기 때문에 복사생성자가 실행될 때 깊은 복사를 수행하도록 만들어져야 할 것 같다.

Dept 소멸자는 객체 내용을 삭제해주는 코드를 구현해야 한다고 생각한다.

Read 함수는 cin 을 통해 점수 10 개의 입력을 받아 score 에 넣어주는 기능을 해주는 함수이다.

isOver60 함수는 scores 에 들어있는 scores 의 배열에 들어있는 값들을 60 과 비교해 bool 값으로 반환해주는 함수이다.

2 번

복사생성자를 제거해도 오류가 발생하지 않도록 코드를 고치기 위해서는 일단 복사생성자를 없애고 countPass 에 매개변수를 Dept 로 넣지 말고, scores 배열을 넣는 방식으로 변경하면 될 것 같다는 생각을 했다

아이디어 평가 :

문제 해결 방법에 작성했던 방식들을 거의 그대로 구현했다.

몇 가지 예상과 다르게 흘러간 부분은 read 함수의 cin 부분이 for 구문을 사용하여 엔터를 치도록 하면 줄 바꿈이 되는 대신 자동으로 다음 부분으로 넘어가는데, space 로 입력을 넣으면 자동으로 넘어가지 않는 부분이었다. 10 개 이상의 입력을 넣어도 앞을 10 개의 숫자만 비교하도록 짜여 있지만, 엔터를 누르지 않는 이상 10 개의 입력이 넘어가도

자동으로 다음 부분으로 넘어가지 않는 부분이 아쉬웠다.

두번째는 2 번 문제에 countPass 함수를 변경할 때 매개변수에 scores 배열을 넣는 방식을 생각하고 있었는데, 이것이 Dept 클래스의 멤버 함수로 넣어버리면 매개변수 없이 활용할 수 있어서 방식을 변경했다.

## 알고리즘 설명

1.

헤더 파일에서는 Dept 클래스에 문제에 나와있는 것과 같이 멤버를 구성하였다.

Int size : 입력 받을 수의 개수를 정하는 변수

Int\* scores : 입력 받는 수를 저장하는 배열 변수

Dept(int size) : Dept 생성자 size 와 scores 를 초기화해준다.

Dept(const Dept& dept) : Dept 복사생성자 size 와 scores 를 Dept 에서 복사해 초기화해준다.

~Dept() : 소멸자 함수

Int getSize() : size 변수를 반환해주는 함수

Void read() : 10 개의 입력을 받아 socres 배열에 저장하는 함수

Bool isOver60(int index) : scores 배열에 저장된 수가 60 을 넘는지 확인하는 함수

Dept.cpp 파일에서는 헤더파일의 클래스 선언부의 구현부이다.

Dept 복사생성자 : for 문을 사용해서 복사하는 객체의 scores 를 복사해 오고, size 는 그냥 this 를 사용해서 복사해 준다.

~dept 소멸자 : scores 변수가 포인터로 되어있어 동적 생성되었기 때문에, delete[] socres 를 사용해 힙영역을 반환해 준다.

Read 함수 : cout 을 통해 정해진 문자를 출력해주고, for 문을 돌면서 cin 을 통해 scores 배열에 수를 하나씩 입력해 준다.

isOver60 함수 : if 문을 사용해 socres[index]를 60 과 비교해 bool 값을 반환해 준다.

## Main.cpp 파일

countPass(Dept dept) 함수 : count int 변수를 선언하고, 복사생성자를 입력으로 받아, for 문을 통해 객체의 size 만큼 반복하며, isOver60 함수를 통해 60 이 넘으면 count 변수를 1 씩 플러스 해준다. 마무리가 되면 count 를 반환해 준다.

Main() 함수 : 객체를 먼저 생성해 주고, 객체.read() 함수를 실행해 주어서 입력을 받은 다음, countPass 함수를 n 변수를 선언해 몇 명이나 60 점을 넘겼는지에 대한 숫자를 넣어준다. 그 다음 정해진 문자를 출력해 준다.

2.

1 과 거의 동일하지만 몇가지만 변경하였다.

먼저 헤더파일에 countPass 함수를 멤버 함수로 넣어 변경해준다.

countPass 함수를 main.cpp 에서 dept.cpp 파일로 옮겨준다.

countPass 함수는 더 이상 getSize() 함수를 사용하지 않고, size 를 직접 사용한다. 또한 isOver60 함수도 dept 객체에서 가져오는 것이 아닌 직접 사용할 수 있다.

반면 getSize() 멤버 함수는 더 이상 사용하는 곳이 없어 삭제해 준다.

Main 함수에서 countPass 함수를 사용하는 부분은 객체의 멤버변수 countPass 함수를 사용하도록 변경해 준다.