

## 소스 구현 설명

### 문제 정의

12. 다음은 학과를 나타내는 Dept 클래스와 이를 활용하는 main()을 보아 준다.

```
class Dept {
    int size; // scores 배열의 크기
    int* scores; // 통학 할당 받은 점수 배열의 주소

public:
    Dept(int size) { // 생성자
        this->size = size;
        scores = new int[size];
    }
    Dept(const Dept& dept); // 복사 생성자
    ~Dept(); // 소멸자
    int getSize() { return size; }
    void read(); // size 만큼 키보드에서 점수를 읽어 scores 배열에 저장
    bool isOver60(int index); // index의 학생의 성적이 60보다 크면 true 리턴
};

int countPass(Dept dept) { // dept 학과에 60점 이상으로 통과하는 학생의 수 리턴
    int count = 0;
    for (int i = 0; i < dept.getSize(); i++) {
        if (dept.isOver60(i)) count++;
    }
    return count;
}

int main() {
    Dept com(10); // 총 10명이 있는 학과 com
    com.read(); // 총 10명의 학생들의 성적을 키보드로부터 읽어 scores 배열에 저장
    int n = countPass(com); // com 학과에 60점 이상으로 통과한 학생의 수를 리턴
    cout << "60점 이상은 " << n << "명";
}
```

(1) main()의 실행 결과가 다음과 같이 되도록 Dept 클래스에 멤버들을 모두 구현하고, 전체 프로그램을 완성하라.

```
10개 점수 입력>> 10 20 30 40 50 60 70 80 90 100
60점 이상은 4명
```

(2) Dept 클래스에 복사 생성자 Dept(const Dept& dept);가 작성되어 있지 않은 경우, 컴파일은 되지만 프로그램 실행 끝에 실행 시간 오류가 발생한다(복사 생성자를 뺀 채 실행해보라). 위의 코드 어느 부분이 실행될 때 복사 생성자가 호출되는지 설명하고, 복사 생성자가 없으면 왜 실행 오류가 발생하는지 설명하라.

(3) Dept 클래스에 복사 생성자를 제거하라. 복사 생성자가 없는 상황에서도 실행 오류가 발생하지 않게 하려면 어느 부분을 수정하면 될까? 극히 일부분의 수정으로 해결된다. 코드를 수정해보라.

- (1) main()의 실행 결과가 다음과 같이 되도록 Dept 클래스에 멤버들을 모두 구현하고, 전체 프로그램을 완성하라.

```
10개 점수 입력>> 10 20 30 40 50 60 70 80 90 100
60점 이상은 4명
```

- (2) Dept 클래스에 복사 생성자를 제거하라. 복사 생성자가 없는 상황에서도 실행 오류가 발생하지 않게 하려면 어느 부분을 수정하면 될까? 극히 일부분의 수정으로 해결된다. 코드를 수정해보라.

## 문제 해결 방법

문제에서 Dept 클래스의 복사 생성자를 제거하고, 오류를 해결하라고 하였기에, Dept 클래스의 복사 생성자를 제거했다.

제거하니 메모리 할당을 해제할 때 에러가 발생했고, 복사 생성자가 사라지며 디폴트 복사 생성자가 사용되어 얇은 복사가 이루어져 메모리 해제 시 동적 메모리 주소를 공유하던 모든 객체가 해제되어 에러가 난 것으로 생각되었다.

때문에 Dept 클래스의 복사 생성자의 기능(깊은 복사)을 유지하기 위해, countPass(Dept dept) 함수의 매개변수들이 각각의 동적 메모리 주소를 얻어 깊은 복사를 유지하도록, 참조(&)매개변수로 설정해주었다.

Dept dept -> Dept& dept

## 아이디어 평가

문제 해결을 위해, 수업 시간에 배운 생성자와 복사 생성자, 얇은 복사와 깊은 복사 개념을 적극적으로 활용하여 수정에 임했다.

결정적으로 참조자를 추가하여 해결했다.

## 문제를 해결한 키 아이디어 또는 알고리즘

### 1. 생성자와 복사 생성자

생성자란, 객체가 생성될 때 호출되는 일종의 초기화 메서드이다. 생성자가 생성되는 시점에 함께 호출되어 객체를 초기화해준다.

복사 생성자는, 객체와 동일한 타입의 객체를 매개변수로 받아 전달받은 객체와 동일하게 객체를 복사하는 생성자이다. 복사 생성자가 없을 경우 디폴트 복사 생성자가 사용되며, 이와 같은 경우 얇은 복사가 이루어진다.

### 2. 참조

참조란, "객체의 다른 이름"으로, 포인터와 같이 어떠한 주소를 담고 그 주소를 통해 객체를 가르키는 것이 아니라, 객체 그 자체를 직접 수정할 수 있는 것입니다. 객체를 가르키는 것이 아니기에 무언가를 담는 변수가 아니고, "객체를 부를 수 있는 또다른 이름" 그 자체라고 볼 수 있습니다.

### 3. 깊은 복사와 얇은 복사

깊은 복사와 얇은 복사는 같은 "복사"라는 역할을 수행하지만, 얇은 복사는 대상이 동적 메모리에 할당된 경우 그 메모리의 주소만 복사하고, 동적 메모리의 주소를 공유합니다. 반대로 깊은 복사는 동적 메모리를 새롭게 할당하여 독립된 객체로 복사하는 완전한 복사입니다. 때문에, 동적 메모리가 반환될 경우 얇은 복사가 이루어진 모든 객체가 동시에 반환되므로 코드에 따라 오류가 발생할 수 있습니다.