

객체지향프로그래밍 과제#3

소프트웨어전공 202284011 김연재

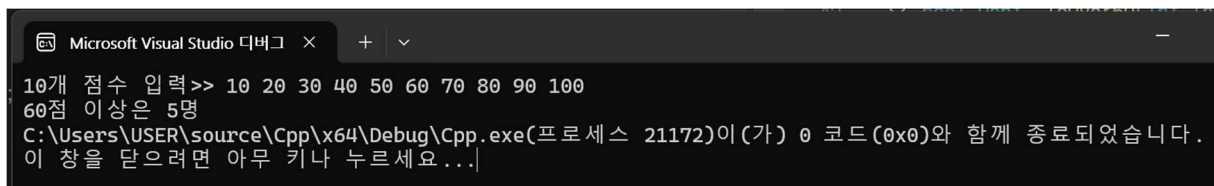
소프트웨어전공 202004006 고창석

- (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 5명

[소스 수행 결과 화면 캡처]



```
Microsoft Visual Studio 디버그 × + -
10개 점수 입력>> 10 20 30 40 50 60 70 80 90 100
60점 이상은 5명
C:\Users\USER\source\Cpp\x64\Debug\Cpp.exe(프로세스 21172)이(가) 0 코드(0x0)와 함께 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요...
```

[소스 구현 설명]

Dept 클래스에 구현한 멤버들은 다음과 같다.

```
Dept::Dept(const Dept& dept) {
    this->size = dept.size;
    this->scores = new int[dept.size];
    for (int i = 0; i < dept.size; i++) {
        this->scores[i] = dept.scores[i];
    }
}
```

복사 생성자. 깊은 복사를 수행하며 원본 객체와 복사된 객체가 각각 독립적인 메모리를 가지도록 하였다. 이 코드에서는 dept.score의 각 값을 새로운 배열 this->scores에 복사하도록 하였다.

```
Dept::~~Dept() {
    if (scores)
        delete[] scores;
}
```

소멸자. delete[] scores;는 new로 할당된 배열을 삭제하여 메모리 누수를 방지한다.

```
void Dept::read() {
    cout << size << "개 점수 입력>> ";
    for (int i = 0; i < size; i++) {
        cin >> scores[i];
    }
}
```

read 함수. 주어진 프로그램을 해결하기 위해 점수를 어떻게 입력받을 것인가에 대해서 고민해 봤을 때, 주어진 코드에 의하면 객체 com.read가 있어서 read함수에서 반복문을 이용하여 입력을 받아야겠다고 판단했다.

size는 10으로 주어졌으므로 size만큼 반복횟수를 정한 for문을 이용하였고 cin으로 입력받았는데, cin은 띄어쓰기를 하면 그 전까지 입력받고 종료되기 때문에 다른 함수들을 사용하지 않고 cin만 반복하면 된다고 생각하였다.

```
bool Dept::isOver60(int index) {
    if (scores[index] >= 60)
        return true;
    else
        return false;
}
```

isOver60 함수. 이 함수는 배열의 인덱스에 있는 점수가 60점 이상인지 확인해서 60점 이상이면 true, 아니면 false를 반환한다.

- (2) 위의 코드 어느 부분이 실행될 때 복사 생성자가 호출되는지 설명하고, 복사 생성자가 없으면 왜 실행 오류가 발생하는지 설명하라.

countPass 함수가 호출될 때 복사 생성자가 호출된다. 이 함수는 Dept 객체를 값으로 전달받기 때문에 함수 내부에서 Dept dept라는 복사본이 생성된다. 이때 복사 생성자가 호출되어 Dept 객체의 깊은 복사를 수행한다.

복사 생성자가 없으면 기본 복사 생성자가 사용된다. 이는 얇은 복사를 수행한다. 얇은 복사에서는 객체의 멤버 변수들이 단순히 복사되는데, Dept 클래스의 경우 scores는 포인터이므로, 포인터 값만 복사되며, 두 객체가 같은 메모리 주소를 참조하게 된다. 원본 객체와 복사본 객체가 같은 scores 배열을 가리키는 상황이 발생한다.

countPass 함수가 종료될 때 복사본 객체가 소멸되고, 이 과정에서 scores 배열의 메모리가 해제된다. 하지만 원본 객체는 여전히 해제된 메모리 주소를 가리키고 있기 때문에, 이후 com 객체가 scores 배열에 접근하려 하면, 이미 해제된 메모리에 접근하는 오류가 발생한다.

- (3) Dept 클래스에 복사 생성자를 제거하라. 복사 생성자가 없는 상황에서도 실행 오류가 발생하지 않게 하려면 어느 부분을 수정하면 될까?

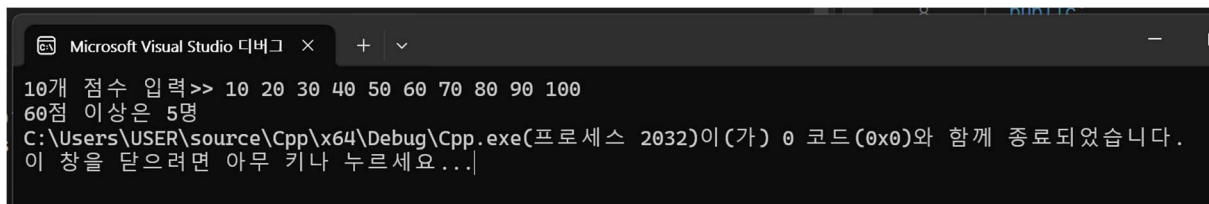
```
int countPass(Department dept) {
    int count = 0;
    for (int i = 0; i < dept.GetSize(); i++) {
        if (dept.isOver60(i))
            count++;
    }
    return count;
}

int countPass(Department& dept) {
    int count = 0;
    for (int i = 0; i < dept.GetSize(); i++) {
        if (dept.isOver60(i))
            count++;
    }
    return count;
}
```

(좌) 원본 코드

(우) 복사 생성자를 제거한 코드

[소스 수행 결과 화면 캡처]



Microsoft Visual Studio 디버그 콘솔 캡처:

```
10개 점수 입력>> 10 20 30 40 50 60 70 80 90 100
60점 이상은 5명
C:\Users\USER\source\Cpp\x64\Debug\Cpp.exe(프로세스 2032)이(가) 0 코드(0x0)와 함께 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요...
```

[소스 구현 설명]

객체 전달 방식을 수정했다. (좌) 코드에서 countPass 함수는 Dept dept로 매개변수를 받는다. 즉 값 전달 방식이다. (우) 코드는 countPass 함수의 매개변수를 참조로 전달하게 수정하였다. 참조로 전달하면 복사본을 만들지 않고 원본 객체를 직접 참조하기 때문에 복사 생성자를 호출하지 않는다.

[평가]

김연재 : 완전 처음 배우는 개념이라 강의만 들을 때는 완전히 이해가 되지 않았는데, 직접 문제 해결 프로그래밍을 해보니 이해가 되었다. 다른 언어에서는 잘 보기 힘든 매력적인 기능인 듯 하다.

고창석 : ppt에 있는 예제 문제를 풀어볼때까지 복사가 깊고 얇은 것에 대해서 이해하지 못했는데 직접 문제를 풀어보면서 오류를 경험해 보니까 복사 생성자를 어떻게 사용하는지, 개념에 대해서 정확히 알게되었고 참조 연산자의 엄청난 편리성도 느꼈다.