

202284023 송성진

객체지향과제#3

소스 수행결과

```
Microsoft Visual Studio 디버그 콘솔

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

C:\Users\User1\source\repos\Project GNU\64\Debug\Project GNU.exe(프로세스 16728)이(가) 0 코
다.
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...
```

이번 과제는 Dept 클래스를 이용하여 10명의 점수를 입력받고, 60점 이상인 사람의 수를 출력하는 프로그램을 구현하고, Dept 클래스에 복사 생성자를 제거하여 어떤 부분을 수정해야 다시 코드가 구현되는지 알아보는 과제입니다.

```
#include <iostream>
using namespace std;

class Dept {
    int size;
    int* scores;
public:
    Dept(int size) {
        this->size = size;
        scores = new int[size];
    }

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

    ~Dept() {
        delete[] scores;
    }

    int getSize() { return size; }

    void read() {
        cout << "10개 점수 입력>> ";
        for (int i = 0; i < size; i++) {
            cin >> scores[i];
        }
    }

    bool isOver60(int index) {
        return scores[index] > 60;
    }
};

int countPass(Dept dept) {
    int count = 0;
    for (int i = 0; i < dept.getSize(); i++) {
        if (dept.isOver60(i)) count++;
    }
    return count;
}

int main() {
    Dept com(10);
    com.read();
    int n = countPass(com);
    cout << "60점 이상은 " << n << "명" << endl;
    return 0;
}
```

위 코드는 주어진 문제를 성공적으로 구현한 코드이며

```
위 코드에서 Dept(const Dept& dept) {  
    size = dept.size;  
    scores = new int[size];  
    for (int i = 0; i < size; i++) {  
        scores[i] = dept.scores[i]; 이 코드를 지워 복사 생성자 제거를 합니다.
```

그 후

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

return count; } 이 코드를 추가하여 Dept 객체를 참조로 전달하도록 변경하여 실행을 해보고, 복사 생성자 없이 코드가 구현되는 것을 확인하였습니다.

문제 해결 방법으로는 동적 메모리 할당과 해제로 메모리 누수를 방지하였고, 깊은 복사를 사용하여 한 객체가 소멸될 때 다른 객체의 메모리도 해제되는 문제를 해결했습니다.

read 함수와 isOver60 함수를 사용하여 scores 배열에 점수를 입력받고, 특정 학생의 점수가 60점 이상인지 여부를 확인하고, 60점 이상인 학생을 세는 기능을 구현하였습니다.

이 프로그램의 알고리즘은 Dept 클래스 생성해 학생들의 점수를 저장하고 관리한 후 생성자에서 배열을 크기만큼 동적으로 할당합니다. read함수에서 사용자로부터 배열에 각 학생의 점수를 입력받고, isOver60 함수로 점수를 확인한 후 결과를 출력하는 알고리즘을 갖게 하였습니다.