

객체지향 프로그래밍 주차과제

202284041최원기, 202284032이종현

C++ 프로그램의 실행 과정과 복사 생성자, 참조 대개 변수에 대한 이해

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

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

276

명품 C++ 프로그래밍

```
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 뻔 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.

위 문제의 정의

위 문제의 정의는 학생들의 성적을 입력, 저장하고 입력된 점수가 60점 이상일 경우의 학생들의 수를 파악하기 위한 코드를 만들어야 한다. 이를 만들기 위해서 dept 클래스 객체를 사용해 점수를 관리하고 countpass를 통해서 60점 이상의 학생수를 파악하여 몇 명인지를 파악하는 것이 주된 문제의 정의라고 할 수 있다.

문제 해결 방법

위의 1번 문제를 해결하기 위해서는 학생의 숫자와 학생들의 점수를 관리하는 dept클래스를 구현하고 활용하는 것이 중요하며 3번 문제 해결을 위해 복사 생성자를 제거하고도 프로그램이 오류가 발생하지 않고 정상적인 작동을 하도록 코드를 수정하는 것이 목표이다.

아이디어 평가

위 문제에 대하여 제출한 코드에서 제시된 아이디어는

1. dept 클래스 관련 메서드를 통한 객체들의 역할을 분리하여서 객체지향적인 설계를하였음.
2. 동적 메모리 할당과 복사 생성자를 사용해 메모리 관리를 체계적으로 구현
3. 참조 전달 방식 사용
4. const 키워드 사용으로 코드의 안정성 증가

1.에 관한 수행 평가 및 결과

수행 평가: 객체의 역할이 잘 분리되어 코드의 구조가 한눈에 알아보기가 쉬워졌다.

결과: 문제 발생시에 간단하게 수정이 가능하므로써 프로그램의 유지보수가 쉬워질듯함.

2.에 관한 수행 평가 및 결과

수행 평가: 동적 메모리 할당과 깊은 복사를 통해 메모리 관리가 잘 되었음, 또한 복사 생성자가 잘 구현되어 메모리 충돌 문제가 발생하지 않음.

결과: 안정적인 메모리 관리를 통해 프로그램이 잘 실행되었음.

3에 관한 수행 평가 및 결과

수행 평가: 참조 전달 방식을 도입하여 불필요한 객체 복사가 줄어들었음

결과: 불필요한 객체 복사가 줄어 처리 속도가 향상되었음

4에 관한 수행 평가 및 결과

수행 평가: const를 사용하여서 안정성을 증가시킴.

결과: 코드의 안정성이 증가됨으로써 오류 가능성이 줄어들음

문제해결에 관한 알고리즘 설명

제일 먼저 프로그램을 시작하면 학생의 수에 해당하는 정수를 입력받아

dept클래스의 객체생성을하고 dept클래스의 read를 호출하여 지정한 수의 점수를 입력받는다. 그 후에 countpass를 호출하여서 점수의 배열을 검사하고 60점 이상의 점수가 몇 명있는지 세기위해 반복문을 사용한다.

사용하여 결과값이 나오게 되면 화면에 "60점 이상인 학생은 00명"으로 표기

학생수를 확인후 프로그램 종료시 dept 클래스 소멸자가 호출되어 할당된 메모리를 해체한다.