

문제 정의

202104255 윤여준

- 위 문제는 키보드로 10명의 학생들의 점수를 입력 받고 60점 이상으로 통과하는 학생들의 수를 계산하는 Dept 클래스를 완성하는 문제입니다.
- Dept 클래스의 멤버 변수와 생성자 및 소멸자, 멤버 함수들이 정의되어 있기 때문에 구현 되어있지 않은 생성자 및 소멸자, 함수를 구현하여 프로그램을 완성할 수 있습니다.

구현 해야 할 생성자 및 소멸자, 함수들은 다음과 같습니다.

- Dept(const Dept& dept); // 복사 생성자
- ~Dept(); // 소멸자
- void read(); // 키보드로 입력받아 배열에 저장
- isOver60(int index); // 60 이상이면 true return

- countPass(Dept dept); // 60 이상인 학생들의 수를 return

그리고 1번과 3번 문제를 해결 하기 위해서는 복사 생성자와 참조 매개 변수에 대한 이해가 필요합니다.

문제 해결 방법

우선 Dept 클래스에 정의 되어 있는 복사 생성자와 소멸자를 구현했습니다.

- 복사 생성자 & 소멸자
 - class를 매개변수로 전달 해야 하기 때문에 깊은 복사 생성자를 구현하였습니다.

그 후 멤버 함수인 `void read();` 와 `bool isOver60();` 를 구현했습니다.

- `void read();`
 - 반복문을 사용하여 Dept 클래스의 scores 배열에 학생 점수를 저장했습니다.
- `isOver60();`
 - if 문을 사용하여 60점 이상인지 판별했습니다.
- `Int countPass(Dept dept);`
 - 최종적으로 구해야 하는 값은 10명의 학생 중 60점이 넘는 학생의 수 이므로 반복문을 사용하여 계산하도록 했습니다.

아이디어 평가

복사 생성자는 강의 시간에 배운 깊은 복사를 구현하기 위해 강의 자료를 참고하여 구현했습니다.

- 복사 생성자

- 깊은 복사 생성자를 구현하였습니다.

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

- 소멸자

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

- void read();

- 반복문을 사용하여 10개의 정수를 입력받고, 동시에 scores 배열에 저장했습니다.

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

- isOver60(int index);
 - index 값을 매개변수로 받아 scores의 배열의 index 번째 값이 60점이 넘으면 true, 넘지 않으면 false를 반환했습니다.

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

- int countPass(Dept dept);
 - for문을 사용하여 Dept 클래스의 scores 배열의 모든 값을 isOver60() 함수로 계산했습니다.

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

문제를 해결한 키 아이디어 또는 알고리즘 설명

이번 문제는 객체를 복사할 때 얕은 복사와 깊은 복사의 차이점을 명확히 인지하고, 복사 생성자를 정확히 알려주는 문제였다고 생각합니다.

기본 생성자, 소멸자, 값을 입력 받고 배열에 저장하는 함수 `void read()`, 60점 이상인지 판별하는 `bool isOver60()` 함수는 이전까지 배운 개념들로 충분히 구현할 수 있었습니다.

허나 `int countPass(Dept dept)` 함수는 객체를 매개변수로 전달하여 그 객체의 `private` 멤버에 접근하는 방식을 사용하는데, 이때 복사 생성자의 역할을 이해하고 복사 생성자와 참조 매개 변수와의 관계를 이해하는 것이 1번과 3번 문제를 해결하는 키 아이디어라고 할 수 있습니다.