STL 과제

2022182024 윤이하나

[ 과제 해결 방법 ]

과제 1.

고민한 내용: 중간고사 때 Bird 객체를 컨테이너에 저장하라는 1번 문제의 답을 똑같이 적으면 해결될 것이라고 생각하여 코드를 작성하였다. 그러나 ''Player::Player(const Player &)': 삭제된 함수를 참조하려고 합니다.'라는 오류가 발생하였다.

|  |  |
| --- | --- |
| 오류 | 조사 내용 |
| 'Player::Player(const Player &)': 삭제된 함수를 참조하려고 합니다. | Player 클래스의 unique\_ptr 객체 때문이다. 해당 객체는 복사 생성이 불가능하다. |

조사한 내용을 바탕으로, 복사 생성자 함수를 추가하거나 move 함수를 추가하여 해결해야겠다고 생각하였다. 그러나 복사 생성자를 추가하였음에도 여전히 같은 오류가 발생하였다.

화면 출력 결과:

출력에 실패하였다.

과제 2.

고민한 내용: 수업 시간 중 numeric\_limits를 이용해 최댓값을 구했던 경험을 토대로 하여 아래 내용과 같이 코드를 작성하면 될 것이라고 생각하였다.

double max = std::numeric\_limits<double>::min();

for (double& num : a)

if (max < num)

max = num;

std::cout << "최댓값: " << max << std::endl;

단, Player의 점수는 큰 값이므로 int가 아닌 double이어야 한다고 판단하였다.

또는 아래와 같이 작성할 수 있다.

auto pos = std::max\_element(a.begin(), a.end());

std::cout << "최댓값: " << \*pos << '\n';

max\_element()를 사용하면 변수의 타입을 명시적으로 저장하지 않아도 된다.

[ 제출한 과제해결 방법이 효율적이라고 주장하는 내용 / 개선할 부분 ]

과제 1.

해결하지는 못하였으나, 복사 생성자를 추가하는 것과 move 함수를 추가하는 것에 있어서 두 가지를 다 해야하는지 아니면 둘 중 한 방법만 사용하여도 되는지 의문을 갖게 되었다.

과제 2.

max\_element()를 사용하면 변수의 타입을 명시적으로 저장하지 않아도 되기 때문에 타입을 지정해줘야 하는 numeric\_limits()보다 max\_element()를 쓰는 편이 더 효율적일 것이라고 판단하였다. 다만 함수가 구체적으로 어떻게 작동하는지, 두 함수가 어떻게 다른지를 잘 알지 못하기 때문에 판단 과정에서 오류가 있을 수 있다고 생각한다.

[ 과제를 하면서 느낀 점 ]

중간고사 때와 마찬가지로 파일을 읽는 첫 번째 문제에서부터 막힐 것이라고 예상하지 못하였다. 2, 3, 4번은 수업 시간이나 중간고사 때 비슷하게나마 해결해본 적이 있는 문제라고 생각하여서 이번 과제를 조금 만만하게 본 것 같아 반성한다. 중간고사를 치르고도 파일을 읽지 못해 1번 이후 과제는 코드로 작성하지 못한 점이 아쉽다.

단순히 수업시간에 배운 내용을 무작정 사용하려고 하였는데, 그래서인지 ‘이 방법이 왜 효율적인가?’에 대한 질문에 답을 하기가 어려웠다. 함수나 코드를 이해하지 못하고 그저 따라쓰는 데에만 급급했기 때문이다. 앞으로는 보다 더 적극적으로 배운 것에 대해 따로 공부해야겠다고 생각한다.