객체지향프로그래밍[02]
202304035 김유진
202104354 지현준

## #문제 정의

두 명의 선수 이름을 입력받은 뒤, 각 선수의 차례마다 이름이 출력되며 <Enter> 키를 누를 경우 랜덤한 3개의 수가 생성하여 출력한다. 이때 생성된 3개의 숫자가 모두 동일하면 해당 차례의 선수의 이름과 함께 승리했다는 메시지를 출력하고 종료되도록 한다. 만약 3개의 숫자가 동일하지 않다면 특정 메시지를 출력하며 다음 선수 차례를 계속 이어간다. 선수의 이름을 관리하는 Player 클래스와 게임을 실행시키는 GamblingGame 클래스로 작성하는 문제다.

## #문제 해결 방법

### [Player.h]

1. name을 공백으로 초기화시키기 위해 생성자를 선언했다.
2. 선수의 이름을 설정하는 함수를 선언했다.
3. 선수의 이름을 반환하는 함수를 선언했다.

### [Player.cpp] : Player 클래스의 구현부 작성

1. Player 클래스를 사용해야 했으며 두 명의 선수 이름을 입력받고 설정하기 위해 이름을 빈 문자열로 초기화(name = "";)하는 생성자를 작성했다.
2. 입력받은 이름으로 설정하기 위해 string 타입인 name을 인자로 받아 이 값을 name으로 설정하는 setName 함수를 작성했다. 이때 객체의 멤버 변수와 매개변수의 이름이 같으므로 구분하기 위해 this 포인터를 사용했다. 즉, this 객체의 name 멤버 변수에 매개변수로 전달된 name 값을 할당했다.
3. 차례가 됐을 때 해당 선수의 이름을 출력하고 3개의 숫자가 동일할 때 승리 메시지("name님 승리!!)를 출력해야 하므로 Player의 string 타입의 name을 반환하도록(return name;) getName()함수를 작성했다.

### [GamblingGame.h]

1. 2명의 선수가 게임을 진행하므로 Player 클래스의 객체 배열(Player playerArr[2];)을 선언했다.
2. 생성자와 게임의 전반적인 게임 진행을 담당하는 play 함수를 선언했다.
3. Player 클래스를 사용하므로 Player 헤더 파일을 include한다.

### [GamblingGame.cpp] : GamblingGame 클래스의 구현부 작성

1. GamblingGame의 생성자

(1) 게임을 시작하는 메시지를 출력하도록 한다.

(2) 이름에 공백이 포함되는 경우도 존재하므로 공백까지 읽어들이는 getline 함수를 사용했고 이를 name에 저장한다.

(3) 입력받은 name을 playerArr의 Player 객체(playerArr[0], playerArr[1])의 setName() 멤버 함수를 호출해 선수 2명의 name을 설정하도록 했다.

2. play 멤버 함수

(1) 게임 종료 조건이 성립되었을 때 while문을 벗어나기 위해 bool 타입 변수인 end를 선언했으며 성립되기 전까지는 while문을 돌아야 하므로 true값을 주었다.

(2) 난수를 생성해야 하므로 난수 생성기를 초기화한다. (seed값이 항상 같으면 난수를 생성할 수 없기 때문에 time(0)을 준다)
(3) 선수가 번갈아 차례를 가져야 하므로 for문을 사용했다.
(4) 차례가 되었을 때와 승리했을 때 해당 차례의 선수 이름을 출력해야 하므로 playerArr의 Player 객체의 getName 멤버 함수를 호출하여 이름을 리턴 받도록 했다.
(5) if문을 사용해 3개의 난수가 같은지 확인하도록 했다.
(6) 성공했을 때, end에 false를 할당해 while문을 벗어나도록 한다.

**[main.cpp]**

1. main 함수에서 GamblingGame 객체를 생성한다.
2. 생성한 GamblingGame 객체에 대해 play() 함수를 호출하여 게임을 실행시킨다.

## #아이디어 평가

1. Player 클래스에서 setName과 getName 함수를 사용하여 각 선수의 이름을 저장하고 출력하는 작업이 원활하게 이루어졌으며, 이를 통해 각 차례마다 해당 선수의 이름을 출력할 수 있었다. 특히, this 포인터를 사용해 멤버 변수와 매개변수를 구분하여 코드를 보다 알기 쉽게 작성할 수 있었다.
2. 두 명의 선수를 Player 클래스에 대한 배열로 관리하고, 차례가 반복되도록 for문을 사용하여 각 선수가 번갈아가며 차례를 가지고 랜덤한 숫자를 출력하는 알고리즘이 깔끔했다. 또, while문의 조건식을 부울타입의 변수로 주면서 승리 조건이 충족될 때 while문을 벗어나도록 설정하고 break를 주어 차례가 넘어가지 않도록 설정한 로직 역시 게임의 흐름을 쉽게 제어할 수 있었다.
3. srand(time(0))와  rand()을 사용하여 매번 새로운 난수가 생성되도록 한 점과  rand()%3으로 난수의 범위를 0, 1, 2 로 제한한 점은 문제의 요구 사항에 부합했다. 이를 3개의 정수 타입의 변수에 생성한 후, if문과 관계연산자(==), 논리연산자(&&)를 사용해 세 개의 숫자가 모두 동일한지 확인하는 로직 또한 문제 요구 사항에 부합했다.

## #문제를 해결한 키 아이디어 또는 알고리즘 설명

while문을 사용하여 게임을 진행하는 알고리즘이 핵심이라고 생각한다. while문을 사용하여 게임 종료 조건이 성립될 때까지 게임을 진행하는 무한루프를 돌도록 하였다. 그 안에서 선수가 번갈아 차례를 가져야 하므로 I의 값을 0부터 1까지 설정하는 for문을 사용했다. i가 0이면 첫 번째 선수의 차례, i가 1이면 두 번째 선수의 차례가 되도록 했다. 각 선수마다 차례가 되었을 때 해당 차례의 선수 이름을 출력해야 하므로 playerArr의 Player객체의 getName 멤버 함수를 호출하여 이름을 리턴받도록 했다. Enter를 입력 받으면 그 다음줄에 랜덤으로 생성된 3개의 정수와 결과를 출력해야 하므로 하나의 문자를 입력받고 Enter를 입력받을 수 있는 함수인 cin.get()을 사용했다. 그 후, 각 선수의 차례마다 0, 1, 2 의 숫자 중 랜덤하게 3개의 숫자를 생성해야 한다. 그러나 rand()는 0부터 32767 사이의 랜덤한 정수를 발생시키므로 rand()를 3으로 나누어주어 나머지 값이 항상 0, 1, 2가 나오도록 하여 랜덤

한 숫자의 나머지 값을 0, 1, 2로 고정시켜 출력했다. if문을 사용하여 만약 n1(첫 번째 랜덤한 정수)==n2(두 번째 랜덤한 정수)이고 n2==n3(세 번째 랜덤한 정수)이면 승리했으므로 해당 차례의 선수 이름을 출력해야 한다. 따라서 playerArr의 Player객체의 getName 멤버 함수를 호출하여 이름을 리턴받도록 했다. end에 false를 할당하고 while문이 중단되도록 했다. 그 후에 break;를 걸어서 for문을 더 이상 돌지 않도록 한다. break를 걸지 않았다면 i=0일 때, 승리한다면 종료해야 하지만 i=1일 때까지 차례가 한번 더 가기 때문에, break를 걸어주어 for문을 벗어나도록 하는 것이다. 만약 n1==n2이고 n2==n3가 false이면 n1, n2, n3 값과 "아쉽군요"를 출력한 후, 계속해서 for문을 돈다.