

과제2 소스코드 설명 - 서신우

문제 정의

문제를 읽고 나서 해결하기 쉽지 않을 것 같은 것들을 정리..

- 배열 안에 있는 두 사람을 번갈아 나타나도록 하는 로직 필요
- 난수를 생성하고 그 데이터들을 main.cpp 로 가져오는 것
- 난수 데이터를 통해 결과를 도출하는 것

문제 해결

- 배열 안에 있는 두 사람을 번갈아 나타나도록 하기 위해서는 index 라는 변수를 생성하고 index 의 값이 0일 때 1을 넣어주고 index 의 값이 1일 때 0을 넣어줌.. 그렇게 하면 players 배열 안에 있는 두 사람이 번갈아 나타나게 됨..
- 난수를 생성하기 위해서 rand() 메소드를 사용..
GamblingGame 클래스의 객체에서 생성한 난수 데이터들을 받기 위한 배열을 생성해 값을 받아옴..
- isJackpot 이라는 변수를 생성하고 난수 데이터 3개를 받고 3개의 값이 같으면 변수의 값을 바꿔줌으로 해서 게임을 끝내고 승리 메시지를 표시..

Code

게임이 끝나기 전까지 계속 반복시키기 위해 while 문을 사용..

만약 한 사람이 승리하게 된다면 isJackpot 변수에 true 값을 넣어줌으로써 루프를 종료..

```
while (!isJackpot)
{
    // ~~~
    // 누군가 승리시
    // isJackpot = true;
}
```

GamblingGame 클래스에서는 난수를 생성하고 또 결과값을 구하는 로직을 구현..

GamblingGame 클래스의 run() 메소드에서 난수 3개와 크기가 4인 배열을 선언.. 배열에

서 앞 3자리는 난수가 들어가고 마지막에는 게임 승리여부에 따라 1(승리) 또는 0 이 들어감..

```
int* GamblingGame::run() {
    int* array = new int[4];

    srand(time(NULL));
    num1 = rand() % 3;
    num2 = rand() % 3;
    num3 = rand() % 3;

    array[0] = num1;
    array[1] = num2;
    array[2] = num3;

    // 승리여부
    if (num1 == num2 && num2 == num3)
    {
        array[3] = 1;
    }
    else
    {
        array[3] = 0;
    }

    return array;
}
```

main 함수에서는 GamblingGame 클래스의 인스턴스를 통해 데이터를 가져오고 또 GamblingGame의 run() 메소드가 리턴한 배열을 통해 승리 여부를 판별..

```
gameResult = game.run();

// 승리여부는 배열의 4번째 자리에 들어있음..
if (gameResult[3] == 1)
{
    message = players[index].name + "님 승리!!";
}
```

```
        isJackpot = true;
    }
    else
    {
        message = "아쉽군요!";
    }
```