

이 코드는 컴퓨터와 사용자가 가위바위보를 하는 게임에 배팅 시스템과 승률을 계산해주는 기능을 추가한 가위바위보 응용 게임 코드입니다.

사용자는 자신의 자금을 입력하고 매 판마다 배팅 금액을 결정해 컴퓨터와 대결합니다.

먼저 승률 계산을 위한 승리 횟수 wins와 전체 게임 수 total, 그리고 배팅 관리 위한 컴퓨터의 자금 comMoney, 사용자의 자금 myMoney, 배팅 금액 betMoney를 전역 변수로 선언합니다.

추가로 저는 이 코드에서 예외 처리를 위해 대부분 무한 반복문으로 코드를 돌리다가 올바른 입력받으면 break로 빠져나오는 형식으로 작성했습니다.

메인 메서드에서 반복문을 사용해 게임을 시작하게 되는데 자신의 초기 자금(myMoney)을 먼저 입력받습니다. 만약 0원 이하로 입력하면 "1원 이상 입력하세요"를 출력하고 다시 입력 받게 합니다. -> continue 사용, 입력 잘 받으면 break로 빠져나감.

그 후 본격적인 게임을 하기 위해 반복문을 한 번 더 사용합니다. 현재 나의 자금과 컴퓨터의 자금(기본 5000원 세팅, 변경 가능)을 출력해주고 컴퓨터의 선택을 getComChoice() 메서드를 이용해 랜덤으로 1~3 중 값을 받아옵니다.

그 후 유저의 가위바위보 선택을 위해 getUserChoice() 메서드를 이용합니다.

getUserChoice 메서드에서는 '가위(1), 바위(2), 보(3), 종료(4)'를 출력하고 유저에게 선택하게 합니다. 이 메서드에서 또한 1~4 사이의 값을 입력하지 않으면 "1~4 사이의 값을 입력하세요"라는 문구를 출력하고 다시 입력받게 합니다. 제대로 입력 받았다면 유저가 선택한 값을 리턴 시킵니다.

이후 메인 메서드에서 만약 리턴된 값이 4이면 게임을 종료하는 것이기 때문에 break를 이용해 반복문을 빠져나오고 게임을 종료시킵니다.

다음 배팅 금액 설정을 위해 setBetting() 메서드를 이용합니다. 이 메서드에서도 예외 처리를 해주어야 하기 때문에 while문으로 작성하였습니다. 먼저 배팅 금액을 입력받고 예외처리를 합니다. 0원 이하이거나 현재 컴퓨터의 자금 또는 자신의 자금보다 높게 설정할 수 없으므로 각 상황에 대해 if문을 이용해 검사 후 자신 또는 컴퓨터의 자금이 현재 얼마로 되어 있는지 알려준 후 다시 입력받게 합니다. 그 후 올바른 값을 입력하면 break로 빠져나옵니다.

다음 승패 판정을 위해 judgeResult() 메서드를 이용합니다. 매개변수로 컴퓨터의 선택 값 com, 유저의 선택값 user를 넘겨줍니다. 판정 메서드 안에서는 먼저 승률 계산을 위해 총 게임 수부터 증가시킵니다. 이 메서드가 호출 되었다는 건 게임을 진행했다는 뜻

이므로 여기서 게임 수를 증가시킵니다. 그 후 if문으로 승, 패, 무를 판단합니다. 만약 이겼을 경우 승률 계산을 위해 승리 수를 카운트 시키고 내 자금에 배팅 머니만큼 증가, 컴퓨터 자금엔 배팅 머니만큼 감소를 시킨 후 "win"을 리턴시킵니다. 반대로 졌을 경우엔 내 자금에선 배팅 머니만큼 빼고 컴퓨터 자금엔 배팅 머니만큼 증가시킨 후 "lose"를 리턴시킵니다. 무승부는 "draw"를 리턴시킵니다.

이후 승패 결과를 알려주기 위한 printResult() 메서드에 컴퓨터의 선택, 유저의 선택, 그리고 승패무 결과 result를 매개변수로 넘겨줍니다.

이 메서드에선 먼저 선택한 가위바위보를 한글로 알려주기 위해 배열을 사용해 선언해줍니다. -> String[] choices = {"가위", "바위", "보"};

그 후 결과 값을 프린트 해주게 되는데 메인 메서드에서 넘어온 com과 user값은 1, 2, 3으로 이루어져 있습니다. 하지만 배열은 인덱스가 0부터 시작하므로 넘어온 값에서 -1를 시켜주고 그 값에 맞는 한글로 된 가위, 바위, 보를 출력해줍니다.

또한 자금 변동을 알려주기 위해 switch-case문을 이용해 이겼을 땐 승리를 알리고 추가된 배팅 금액을 출력해줍니다. 마찬가지로 졌을 땐 패배를 알리고 -1000원 이런 식으로 배팅 금액을 출력해줍니다. 둘 다 아닐 경우 무승부이니 "무승부"라고 출력해 알려줍니다.

이후 자금이 변동이 있었으므로 if문을 사용해 컴퓨터 또는 유저의 자금이 0원이 되었는지 검사합니다. 만약 둘 중 하나라도 0원이 되었을 경우 "게임 종료. 자금이 모두 소진되었습니다"라는 문구 출력과 함께 break로 빠져나옵니다.

승률 계산을 위해 getWinRate() 메서드를 이용합니다. 이 메서드에서는 아까 judgeResult() 메서드에서 이용된 값을 계산합니다. 반올림 계산을 하기 위해 round 함수를 써준 후 total과 wins를 이용해 승률을 리턴시키고 만약 총 게임 수가 0일 시 게임을 안 했으므로 0을 리턴시켜줍니다.

최종적으로 이제 승률과 금액을 출력한 후 게임(코드)을 종료합니다.