

```
자금 입력
===== 가위바위보 게임 =====
초기 자금 입력: 5000

[현재 상태]
내 자금: 5000원
컴퓨터 자금: 5000원

[가위바위보]
가위(1), 바위(2), 보(3), 종료(4):
```

자금 잘못 입력

```
===== 가위바위보 게임 =====
초기 자금 입력: 0
1원 이상 입력하세요.

===== 가위바위보 게임 =====
초기 자금 입력: 5000

[현재 상태]
내 자금: 5000원
컴퓨터 자금: 5000원

[가위바위보]
가위(1), 바위(2), 보(3), 종료(4):
```

가위바위보 잘못 입력

```
[가위바위보]
가위(1), 바위(2), 보(3), 종료(4): 5
1~4 사이의 값을 입력하세요.

[가위바위보]
가위(1), 바위(2), 보(3), 종료(4):
```

베팅 입력 후 승리

```
[가위바위보]
가위(1), 바위(2), 보(3), 종료(4): 2

베팅 금액 입력: 1000

[결과]
컴퓨터: 가위
당신: 바위
=====
승리 +1000원

[현재 상태]
내 자금: 6000원
컴퓨터 자금: 4000원

[가위바위보]
가위(1), 바위(2), 보(3), 종료(4):
```

컴퓨터의 자금을 초과한 베팅

```
[현재 상태]
내 자금: 8000원
컴퓨터 자금: 2000원

[가위바위보]
가위(1), 바위(2), 보(3), 종료(4): 2

베팅 금액 입력: 3000
컴퓨터의 자금이 부족합니다.
컴퓨터의 자금 : 2000원

베팅 금액 입력: |
```

베팅 입력 후 패배

```
[가위바위보]
가위(1), 바위(2), 보(3), 종료(4): 3

베팅 금액 입력: 1000

[결과]
컴퓨터: 가위
당신: 보
=====
패배 -1000원

[현재 상태]
내 자금: 5000원
컴퓨터 자금: 5000원

[가위바위보]
가위(1), 바위(2), 보(3), 종료(4):
```

유저의 자금을 초과한 베팅

```
[현재 상태]
내 자금: 2000원
컴퓨터 자금: 8000원

[가위바위보]
가위(1), 바위(2), 보(3), 종료(4): 3

베팅 금액 입력: 3000
자금이 부족합니다.
현재 자금 : 2000원

베팅 금액 입력:
```

베팅 입력 후 무승부

```
[가위바위보]
가위(1), 바위(2), 보(3), 종료(4): 1

베팅 금액 입력: 1000

[결과]
컴퓨터: 가위
당신: 가위
=====
무승부

[현재 상태]
내 자금: 5000원
컴퓨터 자금: 5000원

[가위바위보]
가위(1), 바위(2), 보(3), 종료(4):
```

컴퓨터 자금 소진 후 승리

```
[현재 상태]
내 자금: 8000원
컴퓨터 자금: 2000원

[가위바위보]
가위(1), 바위(2), 보(3), 종료(4): 1

베팅 금액 입력: 2000

[결과]
컴퓨터: 보
당신: 가위
=====
승리 +2000원

게임 종료
자금이 모두 소진되었습니다.

=====
최종 승률: 75.0%
최종 금액: 10000원
게임을 종료합니다.

종료 코드 0(으)로 완료된 프로세스
```

게임 중도 종료

```
[현재 상태]
내 자금: 7000원
컴퓨터 자금: 3000원
3

[가위바위보]
가위(1), 바위(2), 보(3), 종료(4): 4

=====
최종 승률: 50.0%
최종 금액: 7000원
게임을 종료합니다.

종료 코드 0(으)로 완료된 프로세스
```

내 자금 소진 후 패배

```
[현재 상태]
내 자금: 1000원
컴퓨터 자금: 9000원

[가위바위보]
가위(1), 바위(2), 보(3), 종료(4): 3

베팅 금액 입력: 1000

[결과]
컴퓨터: 가위
당신: 보
=====
패배 -1000원

게임 종료
자금이 모두 소진되었습니다.

=====
최종 승률: 29.0%
최종 금액: 0원
게임을 종료합니다.

종료 코드 0(으)로 완료된 프로세스
```

```

Kjp20211371_Mid2.java
import java.util.*;

public class Kjp20211371_Mid2 {
    // 전역변수
    static int wins = 0; // 승률 계산을 위한 승리 카운팅 2개 사용 위치
    static int total = 0; // 승률 계산을 위한 판수 카운팅 3개 사용 위치
    static int comMoney = 5000; // 컴퓨터의 초기 자금 6개 사용 위치
    static int myMoney; // 유저의 초기 자금 9개 사용 위치
    static int betMoney; // 배팅 금액 10개 사용 위치

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        // 게임 시작
        while(true){
            System.out.println("===== 가위바위보 게임 =====");
            System.out.print("초기 자금 입력: ");
            myMoney = sc.nextInt();
            if(myMoney <= 0){ // 예외 처리
                System.out.println("1원 이상 입력하세요.\n");
                continue;
            }
            break;
        }

        while(true) {
            // 현재 상태 출력
            System.out.println("\n[현재 상태]");
            System.out.println("내 자금: " + myMoney + "원");
            System.out.println("컴퓨터 자금: " + comMoney + "원");

            // 컴퓨터 선택
            int com = getComChoice();
            //System.out.println(com); // 텍스트용 컴퓨터의 선택 출력

            // 가위바위보 선택
            int user = getUserChoice(sc);
            if(user == 4) break; // 4번 선택 시 게임 종료

            // 배팅 금액 입력
            setBetting(sc);

            // 승패 판정
            String result = judgeResult(com, user);

            // 결과 출력
            printResult(com, user, result);

            // 유저 또는 컴퓨터의 자금이 0원이 될 경우 게임 종료
            if(myMoney <= 0 || comMoney <= 0) {
                System.out.println("\n게임 종료\n자금이 모두 소진되었습니다.");
                break;
            }
        }

        // 최종 결과
        System.out.println("\n=====");
        System.out.println("최종 승률: " + getWinRate() + "%");
        System.out.println("최종 금액: " + myMoney + "원");
        System.out.println("게임을 종료합니다.");
        sc.close();
    }
}

```

```

// 컴퓨터 선택
public static int getComChoice(){ 1개 사용 위치
    return (int) ((Math.random() * 3) + 1); // 랜덤 함수 이용
}

// 유저 선택
public static int getUserChoice(Scanner sc){ 1개 사용 위치
    int choice;
    while(true){
        System.out.println("\n[가위바위보]");
        System.out.print("가위(1), 바위(2), 보(3), 종료(4): ");
        choice = sc.nextInt();
        if (choice < 1 || choice > 4) { // 예외 처리
            System.out.println("1~4 사이의 값을 입력하세요.");
            continue;
        }
        break;
    }
    return choice;
}

// 배팅
public static void setBetting(Scanner sc){ 1개 사용 위치

    while(true) {
        System.out.print("\n배팅 금액 입력: ");
        betMoney = sc.nextInt();

        if(betMoney <= 0) { // 예외처리
            System.out.println("0원 이하는 배팅할 수 없습니다.");
            continue;
        }
        // 현재 자금보다 배팅 자금을 높게 설정할 수 없음
        if(betMoney > myMoney) {
            System.out.println("자금이 부족합니다.");
            System.out.println("현재 자금 : " + myMoney + "원");
            continue;
        }

        if(betMoney > comMoney) {
            System.out.println("컴퓨터의 자금이 부족합니다.");
            System.out.println("컴퓨터의 자금 : " + comMoney + "원");
            continue;
        }
        break;
    }
}

```

```

// 승패 판정
public static String judgeResult(int com, int user){ 1개 사용 위치
    total++; // 승률 계산을 위해 총 게임 수 증가

    if(user == com) return "draw"; // 같을 경우 비김
    // 이기는 경우
    if(user == 1 && com == 3 || user == 2 && com == 1 || user == 3 && com == 2){
        wins++; // 승리 수 카운트
        myMoney += betMoney;
        comMoney -= betMoney;
        return "win";
    }
    else { // 지는 경우
        myMoney -= betMoney;
        comMoney += betMoney;
        return "lose";
    }
}

// 결과 출력
public static void printResult(int com, int user, String result){ 1개 사용 위치
    String[] choices = {"가위", "바위", "보"};
    // 현재 choices는 0=가위 1=바위 2=보, 메인에서 넘어오는 값(인덱스)은 +1 되어있으므로 -1 작업 필요함
    System.out.println("\n[결과]");
    System.out.println("컴퓨터: " + choices[com-1]);
    System.out.println("당신: " + choices[user-1]);
    System.out.println("=====");

    switch(result) {
        case "win":
            System.out.println("승리 +" + betMoney + "원");
            break;
        case "lose":
            System.out.println("패배 -" + betMoney + "원");
            break;
        default:
            System.out.println("무승부");
    }
}

// 승률 계산
public static double getWinRate(){ 1개 사용 위치
    if (total == 0) { // 게임 안 했을 시
        return 0;
    } else {
        return Math.round((wins * 100.0) / total); // 반올림 계산
    }
}

```