

PROJECT

1 2 3 4 5
6 7 8 9 10

숫자 야구 게임



박도엽

목차



1 메서드 리뷰

2 DB 연동

3 실행 화면

4 기능 추가

메서드 리뷰

3자리 숫자 랜덤 생성

```
// 3자리 숫자 랜덤 생성 (중복 없는 숫자)
private void generateAnswer() {
    answer = new int[3];
    int i = 0;
    while (i < 3) {
        int num = (int)(Math.random() * 9) + 1; // 1~9
        boolean duplicate = false;
        for (int j = 0; j < i; j++) {
            if (answer[j] == num) {
                duplicate = true;
                break;
            }
        }
        if (!duplicate) {
            answer[i] = num;
            i++;
        }
    }
}
```

스트라이크,볼 체크

```
// 리턴값: int[0]=스트라이크, int[1]=볼
public int[] check(String input) {
    int strike = 0, ball = 0;
    int[] guess = new int[3];
    for (int i = 0; i < 3; i++) {
        guess[i] = input.charAt(i) - '0';
    }
    for (int i = 0; i < 3; i++) {
        if (guess[i] == answer[i]) {
            strike++;
        } else {
            for (int j = 0; j < 3; j++) {
                if (guess[i] == answer[j]) {
                    ball++;
                    break;
                }
            }
        }
    }
    return new int[]{strike, ball};
}
```

게임 기록 저장

```
저장
void saveRecord(String playerName, int attempts) {
    sql = "INSERT INTO game_record (player_name, attempts) VALUES (" + playerName + ", " + attempts + ")";
    try {
        Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);
        PreparedStatement stmt = conn.prepareStatement(sql);

        stmt.setString(1, playerName);
        stmt.setInt(2, attempts);
        stmt.executeUpdate();
        System.out.println("✅ 기록 저장 성공!");
    } catch (SQLException e) {
        System.out.println("❌ 기록 저장 실패: " + e.getMessage());
    }
}
```

DB 연동

```
기록 저장
static void saveRecord(String playerName, int attempts, int digitCount) {
    String sql = "INSERT INTO game_record (player_name, attempts, digit_count) VALUES (?, ?)";
    try {
        Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);
        PreparedStatement stmt = conn.prepareStatement(sql);
        stmt.setString(1, playerName);
        stmt.setInt(2, attempts);
        stmt.setInt(3, digitCount);
        stmt.executeUpdate();
        System.out.println("✅ 기록 저장 성공!");
    } catch (SQLException e) {
        System.out.println("❌ 기록 저장 실패: " + e.getMessage());
    }
}
```

DB 저장 메서드

1

```
1 • SELECT player_name, attempts, play_date
2   FROM game_record
3   ORDER BY attempts ASC, play_date ASC
4   LIMIT 5;
5 • ALTER TABLE game_record ADD COLUMN digit_count INT DEFAULT 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	player_name	attempts	play_date
▶	123	5	2025-06-27 09:08:31
	123	6	2025-06-27 09:18:33
	류현진	7	2025-06-27 09:19:20
	박도엽	8	2025-06-27 09:07:47

쿼리로 랭킹데이터출력 확인

2

🏆 3자리 랭킹 TOP5 🏆

1등: 123 - 시도: 5, 시간: 2025-06-27 09:08:31

2등: 123 - 시도: 6, 시간: 2025-06-27 09:18:33

3등: 류현진 - 시도: 7, 시간: 2025-06-27 09:19:20

4등: 박도엽 - 시도: 8, 시간: 2025-06-27 09:07:47

콘솔창 랭킹 화면 출력

3

실행 화면

```
플레이어 이름 입력: |
```

시작 화면

메인클래스를 실행하고 이름(닉네임) 입력시 게임이 시작됩니다.

1

```
Strike = 1   Ball = 0
숫자 3자리 입력(1~9) : 456
Strike = 1   Ball = 1
숫자 3자리 입력(1~9) : 765
Strike = 1   Ball = 1
숫자 3자리 입력(1~9) : |
```

게임 플레이

숫자 3~4자리(1~9)를 입력하면 룰에 따라 스트라이크,볼이 출력됩니다.

2

```
Strike = 3   Ball = 0
🚩 정답입니다!
🎯 시도 횟수: 7
✅ 기록 저장 성공!

🏆 랭킹 TOP5 🏆
1등: 123 - 시도횟수: 2, 기록시간: 2025-06-25 14:06:45.0
2등: 테스트 - 시도횟수: 3, 기록시간: 2025-06-25 12:31:32.0
3등: 333 - 시도횟수: 4, 기록시간: 2025-06-25 13:28:15.0
4등: 123 - 시도횟수: 4, 기록시간: 2025-06-25 16:18:56.0
5등: 박도연 - 시도횟수: 5, 기록시간: 2025-06-25 15:06:30.0
```

종료 화면

정답을 맞추면 시도 횟수,이름이 DB에 등록되고 시도횟수가 상위 5등 이내면 랭킹에 등록되어 이름, 시도횟수, 기록달성시간이 출력됩니다.

3

기능 추가

난이도를 선택하세요:

1. Easy (3자리)

2. Hard (4자리)

선택 (1 또는 2): 1

숫자 3자리 입력(1~9) : 123

난이도 설정

EASY(3자리) , HARD(4자리) 설정가능

1

Strike = 1 Ball = 1

숫자 3자리 입력(1~9) : 679

Strike = 1 Ball = 2

숫자 3자리 입력(1~9) :

🕒 제한 시간 초과! Game Over!

시간 제한

90초 동안 답을 못맞추면 자동으로 게임종료

2

🏆 3자리 랭킹 TOP5 🏆

1등: 123 - 시도: 5, 시간: 2025-06-26 15:45:37.0

🗨 다시 시작하시겠습니까? (Y/N): Y

🚫 랭킹을 초기화하시겠습니까? (Y/N): Y

🗑 전체 기록이 초기화되었습니다.

플레이어 이름 입력: |

랭킹 초기화

오래된 기록들 삭제

3



Thank you