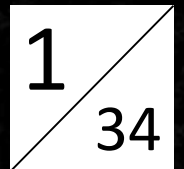


# PROJECT BINGO GAME

15/03/2024



# 목차

날짜

2024.03.15

PROJECT

BINGO GAME

1

팀 소개

2

목표

3

요구사항 정의

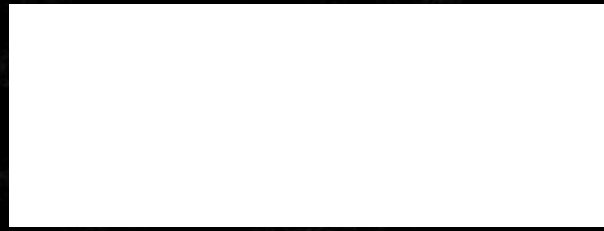
4

코드 분석

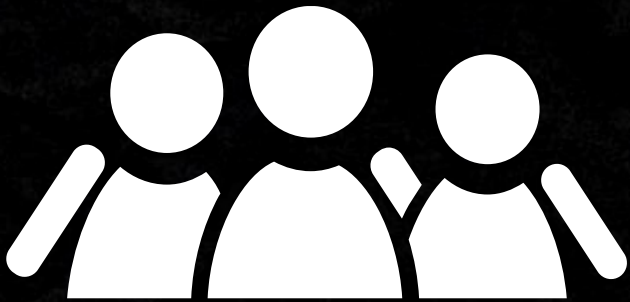
5

경쟁력&기대효과

# 1) 팀 소개



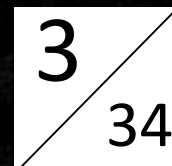
## PROJECT TEAM



조장-전민정

조원-이태형  
이찬윤  
윤성도

TEAM  
NAME  
석빙고





## 2) 목 표

# WHAT WE DO?

Eclipse를 사용하여 3x3빙고 게임을 제작하였습니다.  
위 게임은 3x3빙고칸안에 9개의 퀴즈가 랜덤으로 출력  
되며,출력된 퀴즈의 정답을 맞추면 빙고칸이 한 개 사라  
지며 열과 행, 대각이 연속적으로 3칸 있을시 1BINGO가  
되며 총 3개의 BINGO가 되었을경우 게임이 종료됩니다.

# 3) 요구 사항

요구사항 정의서		
요구사항	내 용	해결 방법
게임의 룰 확립	열 혹은 행으로 3개가 될 경우 빙고라고 떠야한다.	일렬로 3개의 정답을 맞춘 경우 빙고를 카운트 한다.
	대각선으로 'x' 가 3개가 될 경우에도 빙고라고 떠야한다.	대각선으로 3개의 정답을 맞춘 경우 빙고를 카운트 한다.
	3빙고가 되었을경우 게임이 종료되어야한다.	게임의 종료조건을 3빙고를 맞추었을 경우로 지정한다.
부가적 기능	차별성을 위해 히든 카드를 선택하여 특별한 기능을 추가한다.	히든카드의 위치는 랜덤이며,맞출경우 하나를 추가 제거한다.
	매 게임마다 퀴즈가 랜덤으로 나와야 한다.	코드를 작성하여 매 게임마다 퀴즈는 랜덤으로 나오게 한다.
	중복된 퀴즈가 나오지 않게 한다.	이미 낸 문제는 다시 나오지 않게 코드를 작성한다.
	라이프 시스템을 도입하여 게임의 긴장성을 확립한다.	특정 조건을 만들어 틀릴 경우 라이프를 줄어들게 하여 소진시 종료한다.



# 요 구 사 항

## ① 게 임 룰 확 립 해 결 방 법

1. 열 혹은 행으로 3개가 될 경우 빙고라고 한다.

```
for(int i=1;i<6;i+=2) {  
    for(int j=1; j<6 ;j+=2) {  
        if(board[i][j]=='x')  
            bcnt++;  
    }  
    if(bcnt==3) {  
        bingo++;  
    }  
    bcnt=0;  
  
    //세로  
    for(int j=1;j<6; j+=2) {  
        if(board[j][i]=='x') {  
            bcnt++;  
        }  
    }  
    if(bcnt==3) {  
        bingo++;  
    }  
    bcnt=0;  
}
```

가로 혹은 세로로 3개의  
정답을 맞출경우 카운트  
하여

그 카운트가 3개가  
연속적으로 입력 될 경우  
빙고라고 출력한다.

# 요 구 사 항

## ② 게 임 룰 확 립 해 결 방 법

2.대각선으로 3개가 될 경우 빙고라고 한다.

```
        for(int j=i; j<=i; j++) {
            if(board[i][j]=='x') {
                bcnt++;
            }
        }
        if(bcnt==3) {
            bingo++;
        }
    } //왼>오 대각선
    bcnt=0;

    for(int i=1; i<6; i+=2) {
        for(int j=6-i; j>=6-i; j--) {
            if(board[i][j]=='x') {
                bcnt++;
            }
        }
        if(bcnt==3) {
            bingo++;
        }
    } //오<왼 대각선
```

왼쪽에서 대각선  
그리고

우측에서 대각선으로  
3개의 정답을 맞췄을  
경우 빙고라고 출력 한다.

# 요 구 사 항

## ③ 게 임 룰 확 립 해 결 방 법

3.빙고가 3빙고가 되었을 경우 게임을 종료한다.

```
if(bingo==1) {  
  
System.out.println("1 BINGO");  
  
bingo=0;  
  
}else if(bingo==2) {  
  
System.out.println("2 BINGO");  
  
bingo=0;  
  
}else if(bingo>=3) {  
  
System.out.println("3 BINGO");  
  
System.out.println("★WIN★");  
  
break;
```

빙고를 카운트하여  
빙고라는 변수에  
저장한다.

빙고가 3빙고가 되었을  
경우 게임은 종료되며  
WIN글자를 출력한다.

}



# 요 구 사 항

## ① 부 가 적 기 능

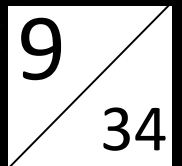
## 해 결 방 법

1. 차별성을 위해 히든 카드를 만들어 특별한 기능을 추가 한다

```
if(usernum==hiddencard) {  
  
if(useranswer.equals(printanswer[hiddencard-1])){  
  
System.out.println("★히든카드를 획득하셨습니다★,  
남아있는 숫자 중 하나를 추가로 지울 수 있습니다!");  
System.out.println("숫자를 선택하세요");  
    int hiddennumber=in.nextInt();  
    in.nextLine();  
    if(hiddennumber<=3) {  
  
        board[1][2*hiddennumber-1]='x';  
    }else if(  
hiddennumber>3&& hiddennumber<=6) {  
  
        board[3][2*hiddennumber-7]='x';  
    }else  
if(hiddennumber>6) {  
  
        board[5][2*hiddennumber-13]='x';  
  
    }  
}
```

히든 카드는 랜덤으로  
들어 있으며,

그 히든 카드를 발견시  
사용자는 원하는 지점의  
문제를 제거 할 수 있다.



# 요 구 사 항

## ② 부 가 적 기 능 해 결 방 법

2.매 게임마다 문제는 랜덤으로 나와야 한다.

```
Random r= new Random();
```

```
String[]printquiz=new String[9];//문제  
9개는 printquiz에서 가져옴
```

```
String[]printanswer=new String[9];
```

```
for(int i=0; i<printquiz.length; i++) {
```

```
int a=r.nextInt(quiz.length);//랜덤으로  
뽑아서 변수 a에 저장
```

```
String randomquiz=quiz[a];
```

문제 배열의 문제를 다수  
확보하여  
매 게임마다 다양한  
퀴즈를 출력 할 수  
있게끔 하였습니다.

# 요 구 사 항

## ③ 부 가 적 기 능 해 결 방 법

3. 중복된 게임이 나오지 않게 한다.

```
printquiz[i]=randomquiz;  
  
printanswer[i]=answer[a];  
  
for(int j=0; j<i; j++) {  
    if(printquiz[i]==printquiz[j]) {  
        i--;  
        break;  
    }  
}
```

중복된 문제가 나올 경우  
뒤로 돌리고 다시 뽑아

중복되지 않은 문제를  
출력시킨다.



# 요 구 사 항

## ④ 부 가 적 기 능 해 결 방 법

4.라이프 시스템을 도입하여 긴장감을 고조시킨다.

```
}else {  
  
System.out.println("오 답!");  
  
life--;  
  
System.out.println("♥x"+life);  
  
}
```

정답을 맞추지 못한 경우  
“오 답” 이라는 문구를  
출력 한다.

그 후 라이프 변수에  
저장 해둔 라이프를  
마이너스 시키고  
출력되는 ♥를 감소  
시킨다.



# 요 구 사 항

## 그 외

## 추 가 기 능

### 빙고 판 제작

```
char[][] board =new char[7][7];
for(int i=0; i<=6;i++) {
for(int j=0;j<=6;j++) {
if(i==0&&j==0) {board[i][j]='┐';
}else if(i==6&&j==6) {board[i][j]='└';
}else if(i==6&&j==0) {board[i][j]='└';
}else if(i==0&&j==6) {board[i][j]='┐';
}else if(i==6&&j==4) {board[i][j]='└┐';
}else if(i==6&&j==2) {board[i][j]='└┐';
}else if(i==4&&j==6) {board[i][j]='┐└';
}else if(i==2&&j==6) {board[i][j]='┐└';
}else if(i==2&&j==0) {board[i][j]='┐└';
}else if(i==4&&j==0) {board[i][j]='┐└';
}else if(i==0&&j==4) {board[i][j]='└┐';
}else if(i==0&&j==2) {board[i][j]='└┐';
}else if(i==2&&j==2) {board[i][j]='┐└';
}else if(i==4&&j==2) {board[i][j]='┐└';
}else if(i==2&&j==4) {board[i][j]='┐└';
}else if(i==4&&j==4) {board[i][j]='┐└';
}else if(i%2!=0&&j==0) {board[i][j]='|';
}else if(i%2!=0&&j==2) {board[i][j]='|';
}else if(i%2!=0&&j==4) {board[i][j]='|';
}else if(i%2!=0&&j==6) {board[i][j]='|';
}else if(i%2==0&&j==1) {board[i][j]='-';
```

```
}else if(i%2==0&&j==5)
{board[i][j]='-';
}
}
}
char num=49;
for(int i=1; i<6; i+=2) {
for(int j=1; j<6; j+=2) {
board[i][j]=num;
num++;
}
```

빙고판의 틀을 제작한다.  
그 후

빙고판 안에 숫자를  
대입하여 빙고판 각 각의  
개인 번호를 가지게 한다.

# Code 분석

## 빙고판 코드

```
char[][] board = new char[7][7];
for(int i=0; i<=6;i++) {
    for(int j=0;j<=6;j++) {
        if(i==0&&j==0) {
            board[i][j]='r';
        }else if(i==6&&j==6) {
            board[i][j]='l';
        }else if(i==6&&j==0) {
            board[i][j]='l';
        }else if(i==0&&j==6) {
            board[i][j]='r';
        }else if(i==6&&j==4) {
            board[i][j]='l';
        }else if(i==6&&j==2) {
            board[i][j]='l';
        }else if(i==4&&j==6) {
            board[i][j]='l';
        }else if(i==2&&j==6) {
            board[i][j]='l';
        }else if(i==2&&j==0) {
            board[i][j]='l';
        }else if(i==4&&j==0) {
            board[i][j]='l';
        }else if(i==0&&j==4) {
            board[i][j]='T';
        }
    }
}
```

```
else if(i==0&&j==2) {
    board[i][j]='T';
}else if(i==2&&j==2) {
    board[i][j]='+';
}else if(i==4&&j==2) {
    board[i][j]='+';
}else if(i==2&&j==4) {
    board[i][j]='+';
}else if(i==4&&j==4) {
    board[i][j]='+';
}else if(i%2!=0&&j==0) {
    board[i][j]='|';
}else if(i%2!=0&&j==2) {
    board[i][j]='|';
}else if(i%2!=0&&j==4) {
    board[i][j]='|';
}else if(i%2!=0&&j==6) {
    board[i][j]='|';
}else if(i%2==0&&j==1) {
    board[i][j]='-';
}else if(i%2==0&&j==3) {
    board[i][j]='-';
}else if(i%2==0&&j==5) {
    board[i][j]='-';
}
}
```

# Code 분석

## 빙고판 코드 - 자료 설명

(1) 자료형 char인 변수 board 선언, 길이는 7, board[0]의 길이는 7으로 2차원 배열을 선언하였다.

(2) 반복문을 통하여 격자무늬인 빙고판의 구조를 만들었다.

```
char num=49;
for(int i=1; i<6; i+=2) {
    for(int j=1; j<6; j+=2) {
        board[i][j]=num;
        num++;
    }
}
```

1	2	3
4	5	6
7	8	9

(1) 빙고판의 구조를 채우고 난 뒤 null값은 1부터 9까지의 빙고 숫자를 채운다.

(2) 자료형을 char로 선언하여 자료형 int는 넣을 수 없으므로 아스키 코드를 활용하여 문자 1에 해당하는 10진수 숫자를 대입하여 아스키코드로 변환 하였다.

(3) 반복문을 통해 문자 1부터 9까지의 값을 대입하였다.

# Code 분석

---



## 퀴즈 & 정답 배열

---

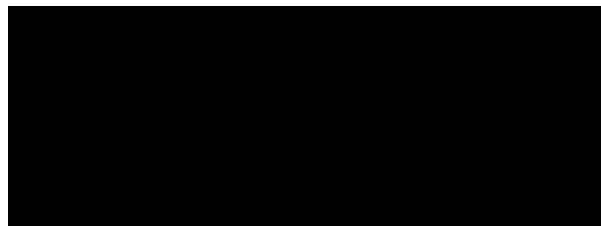
```
String []quiz=new String[15];  
String []answer=new String[15];
```

```
quiz[0]="10원짜리 동전 앞 면에 새겨진 탑의 이름은?";  
quiz[1]="캐나다의 수도는?";  
quiz[2]="자바는 □□지향 프로그래밍이다.";  
quiz[3]="전구를 발명한 사람은?";  
quiz[4]="고래 싸움에 □□ 등 터진다.";  
quiz[5]="산소, 이산화탄소, 질소, 수소 중 가장 가벼운 기체는?";  
quiz[6]="한국에서 대구, 대전, 광주 중 인구가 제일 많은 도시는?";  
quiz[7]="조선 초대 군주의 이름은?";  
quiz[8]="22+2+2+2=?";  
quiz[9]="대한제국의 마지막 황제는?";  
quiz[10]="※폭탄문제※ 오라클의 창업자 이름은?";  
quiz[11]="삼성의 창업자 이름은?";  
quiz[12]="스페인의 수도는?";  
quiz[13]="태양-지구-달 위치로 배열될때 일어나는 현상은?";  
quiz[14]="※폭탄문제※ 한국오라클 사무소 본사는 서울 □□구에  
있다.";
```



# Code 분석

---



## 퀴즈 & 정답 배열

---

```
answer[0]="다보탑";
answer[1]="오타와";
answer[2]="객체";
answer[3]="에디슨";
answer[4]="새우";
answer[5]="수소";
answer[6]="대구";
answer[7]="이성계";
answer[8]="28";
answer[9]="순종";
answer[10]="래리 엘리슨";
answer[11]="이병철";
answer[12]="마드리드";
answer[13]="월식";
answer[14]="강남";
```

# Code 분석

---



## 코드 세부 내용

---

```
Random r= new Random();
String[] printquiz=new String[9];
String[] printanswer=new String[9];
for(int i=0; i<printquiz.length; i++) {
    int a=r.nextInt(quiz.length);
    String randomquiz=quiz[a];
    printquiz[i]=randomquiz;
    printanswer[i]=answer[a];
    for(int j=0; j<i; j++) {
        if(printquiz[i]==printquiz[j]) {
            i--;
            break;
        }
    }
}
```

- (1) 랜덤 클래스를 선언한다.
- (2) 자료형이 string이고 길이는 9인 배열 printquiz와 printanswer을 선언하였다. 실제 빙고에 사용될 배열이다.
- (3) 반복문을 통하여 랜덤으로 범위는 0~14, quiz의 인덱스 번호를 랜덤으로 뽑는다.
- (4) 그 값을 printquiz와 printanswer에 대입한다.
- (5) 중복이 있으면 이전 값과 비교하여 i--; 시켜 한번 더 뽑는다.



# Code 분석

---



## 코드 세부 내용

---

```
int life=3;
Scanner in=new Scanner(System.in);
System.out.println("빙고게임에 오신 것을 환영합니다!");
System.out.println("-----");
System.out.println("번호를 선택하고 문제의 정답을 맞추면 해당 번호를  
지웁니다.");
System.out.println("빙고 3줄을 맞추면 게임이 종료됩니다.");
System.out.println("기회는 ♥x"+ life+"입니다.");
System.out.println("-----");

for (int i = 0; i < board.length; i++) {
    for (int j = 0; j < board[0].length; j++) {
        System.out.print(board[i][j]);
    }
    System.out.println();
}
```

- (1) Life변수 선언, 초기값을 3으로 설정하여 오답의 기회는 3번임을 나타낸다.
- (2) Scanner 클래스 선언하여 키보드로 입력받는다.
- (3) 시작하기 전에 빙고판을 사용자에게 출력한다.

# Code 분석

---



## 코드 세부 내용

---

```
int bingo=0;  
int hiddencard=r.nextInt(9)+1;
```

- (1) Bingo의 개수를 판별한 변수 bingo를 선언한다.
- (2) Hiddencard 변수를 무한 반복문 전에 미리 선언하여 저장한다.  
범위는 1부터 9



# Code 분석

---



## 코드 세부 내용

---

```
for(;;) {
    System.out.println("1부터 9까지의 번호중 하나를
    선택하세요");
    System.out.println("-----");

    int usernum=in.nextInt();
    in.nextLine();
    int []quiznum= {1,2,3,4,5,6,7,8,9};
    if(usernum <=0 || usernum>9) {
        for (int i = 0; i < board.length; i++) {
            for (int j = 0; j < board[0].length;
j++) {
                System.out.print(board[i][j]);
            }
            System.out.println();
        }
        continue;
    }
}
```

- (1) 무한 반복문 실행
- (2) Usernum 변수를 선언하여 사용자에게 숫자를 입력받는다.
- (3) Quiznum 변수를 선언하여 사용자가 입력한 값이랑 quiznum 값을 대조할 것이다.
- (3) 사용자가 1~9 이외의 숫자를 입력할시 반복문은 처음부터 다시 실행된다.

# Code 분석

---



## 코드 세부 내용

---

```
for(int i=0; i<quiznum.length; i++) {  
    if(usernum==quiznum[i]) {  
        System.out.println(printquiz[i]);  
    }  
}
```

(1) 사용자가 입력한 번호에 해당하는 문제가 출력된다

# Code 분석

## 코드 세부 내용

```
String useranswer;
useranswer=in.nextLine();
for(int i=0; i<quiznum.length; i++) {
    if(usernum==quiznum[i]) {
        if(i<3) {
            if(useranswer.equals(printanswer[i])){
                System.out.println("정답!");
                board[1][2*i+1]='x';
            }else {
                System.out.println("오답!");
                life--;
                System.out.println("♥x"+Life);
            }
        }else if(i>=3&&i<6) {
            if(useranswer.equals(printanswer[i])){
                System.out.println("정답!");
                board[3][2*usernum-7]='x';
            }else {
                System.out.println("오답!");
                life--;
                System.out.println("♥x"+Life);
            }
        }else if(i>=6) {
            if(useranswer.equals(printanswer[i])){
                System.out.println("정답!");
                board[5][2*usernum-13]='x';
            }else {
                System.out.println("오답!");
                life--;
                System.out.println("♥x"+Life);
            }
        }
    }
}
```

- (1) 사용자에게 문제의 답을 입력받는다.
- (2) 입력받은 답이 정답이면 정답이라고 출력, 빙고판의 해당 번호의 인덱스 값을 x로 바꾼다.
- (3) 오답이면 오답이라고 출력, life가 1씩 줄어든다.
- (4) usernum값에 따라 'x'로 바꿀 좌표를 반복문으로 처리하였다

# Code 분석

## 코드 세부 내용

```
int bcnt=0;
for(int i=1;i<6;i+=2) {
    for(int j=1; j<6 ;j+=2) {
        if(board[i][j]=='x') {//가로
            bcnt++;
        }
    }
    if(bcnt==3) {
        bingo++;
    }
    bcnt=0;
    for(int j=1;j<6; j+=2) {
        if(board[j][i]=='x') {
            bcnt++;
        }
    }
    if(bcnt==3) {
        bingo++;
    }
    bcnt=0;
}
```

- (1) 반복문을 통해 가로, 세로, 대각선 2개의 빙고 판별기능을 만들었다.
- (2) 가로 반복문은 가로 한줄 빙고 판별하고 변수bcnt를 0으로 초기화시킨다.
- (3) 세로 반복문 또한 세로 한 줄 빙고 판별하고 변수 bcnt를 0으로 초기화시킨다.



# Code 분석

## 코드 세부 내용

```
for(int i=1; i<6; i+=2) {
    for(int j=i; j<=i; j++) {
        if(board[i][j]=='x') {
            bcnt++;
        }
    }
    if(bcnt==3) {
        bingo++;
    }
} //왼>오 대각선
bcnt=0;

for(int i=1; i<6; i+=2) {
    for(int j=6-i; j>=6-i; j--) {
        if(board[i][j]=='x') {
            bcnt++;
        }
    }
    if(bcnt==3) {
        bingo++;
    }
} //오<왼 대각선
bcnt=0;
```

- (1) 대각선 왼쪽>오른쪽 판별이 끝나면 변수 bcnt를 0으로 초기화시킨다.
- (2) 대각선 오른쪽>왼쪽 판별이 끝나면 변수 bcnt를 0으로 초기화시킨다.

# Code 분석

---



## 코드 세부 내용

---

```
for (int i = 0; i < board.length; i++) {  
    for (int j = 0; j < board[0].length; j++) {  
        System.out.print(board[i][j]);  
    }  
    System.out.println();  
}
```

(1)보드판을 출력시킨다

```
if(bingo==1) {  
    System.out.println("1 BINGO");  
    bingo=0;  
}else if(bingo==2) {  
    System.out.println("2 BINGO");  
    bingo=0;  
}else if(bingo>=3) {  
    System.out.println("3 BINGO");  
    System.out.println("★WIN★");  
    break;  
}
```

(1) Bingo값에 따라 n BINGO 출력한다.

(2) Bingo가 3이상이면 break; 기능을 통해 게임이 종료, 즉 무한 반복문이 종료된다.

# Code 분석

---



## 코드 세부 내용

---

```
if(life==0) {  
    System.out.println("재도전하세요");  
    break;  
}
```

(1) Life가 0이되면 재도전하라는 문구가 출력되고 게임종료, 무한 반복문이 종료된다.

# Code 분석

---



## 코드 세부 내용

---

```
if(usernum==hiddencard) {  
    if(useranswer.equals(printanswer[hiddencard-  
1])){  
        System.out.println("★히든카드를  
획득하셨습니다★, 남아있는 숫자 중 하나를 추가로 지울 수  
있습니다!");  
        System.out.println("숫자를 선택하세요");  
        int hiddennumber=in.nextInt();  
        in.nextLine();  
        if(hiddennumber<=3) {  
            board[1][2*hiddennumber-1]='x';  
        }else if(hiddennumber>3&& hiddennumber<=6) {  
            board[3][2*hiddennumber-7]='x';  
        }else if(hiddennumber>6) {  
            Board[5][2*hiddennumber-13]='x';  
        }  
    }  
}
```

- (1) 빙고판별까지 끝난 후 만약 사용자가 입력한 숫자가 hiddencard 숫자였다면, 히든카드를 획득하였다는 문구를 출력하고 사용자에게 숫자를 추가로 입력받는다.
- (2) 사용자가 선택한 번호를 x로 바꾼다.

# Code 분석

---



## 코드 세부 내용

---

```
for(int i=1;i<6;i+=2) {
    for(int j=1; j<6 ;j+=2) {
        if(board[i][j]=='x') {//가로
            bcnt++;
        }
    }if(bcnt==3) {
        bingo++;
    }
    bcnt=0;

//세로
    for(int j=1;j<6; j+=2) {
        if(board[j][i]=='x') {
            bcnt++;
        }
    }
    if(bcnt==3) {
        bingo++;
    }
    bcnt=0;
}
```

(1)빙고를 판별한다.

# Code 분석

---



## 코드 세부 내용

---

```
for(int i=1; i<6; i+=2) {
    for(int j=i; j<=i; j++) {
        if(board[i][j]=='x') {
            bcnt++;
        }
    }
    if(bcnt==3) {
        bingo++;
    }
} //왼>오 대각선
bcnt=0;

for(int i=1; i<6; i+=2) {
    for(int j=6-i; j>=6-i; j--) {
        if(board[i][j]=='x') {
            bcnt++;
        }
    }
    if(bcnt==3) {
        bingo++;
    }
} //오<왼 대각선
bcnt=0;
```

(1)빙고를 판별한다.



# Code 분석

---



## 코드 세부 내용

---

```
for (int i = 0; i < board.length; i++) {  
    for (int j = 0; j < board[0].length; j++) {  
        System.out.print(board[i][j]);  
    }  
    System.out.println();  
}
```

(1) 빙고판과 함께 빙고여부를 출력한다

```
    if(bingo==1) {  
        System.out.println("1 BINGO");  
        bingo=0;  
    }else if(bingo==2) {  
        System.out.println("2 BINGO");  
        bingo=0;  
    }else if(bingo>=3) {  
        System.out.println("3 BINGO");  
        System.out.println("★WIN★");  
        break;  
    }  
} //두 번째 if문 끝  
} //첫번째 if문 끝  
} //무한 for문 끝
```

(1)빙고를 판별한다.

# Effect

---



## 기대 효과

---

각 반복문의 기능을 나누었다.

- 랜덤으로 뽑아서 새 배열에 저장하는 기능
- 문제를 출력하는 기능
- 정답과 오답을 판별하는 기능
- 빙고를 판별하는 기능
- 빙고 여부를 출력하는기능

# Competitive

---



## 경쟁력

---

- 히든카드를 배치함으로서 타 빙고판과는 유사점을 달리할 수 있습니다.
- 그 외에 문제는 각각 랜덤으로 출력되기때문에 매 게임마다 새로운 게임을 즐기는 듯한 느낌을 줄 수 있습니다.
- '라이프'기능을 탑재함으로서 게임을 보다 긴장감 있게 즐길 수 있으며 보다 익스트림한 게임을 즐길 수 있습니다.

THANK YOU

---

TEAM

석빙고