



타짜게임 프로젝트

이찬 | 박진우 | 이원재

목 차

- 1 주제 및 기능 선정
- 2 기능별 알고리즘 소개
- 3 실행 화면
- 4 후기

주제

게임 주제 선정
“단계별 타자 게임”

기능

1

게임 시작/종료 및 단계 설정

2

2차원배열에 담긴 단어를 랜덤으로 사용

3

정답/오답에 따른 점수 및 정확도 계산

4

n단계 종료에 따른 게임 진행 선택지

5

실력(점수/정확도)에 따른 빠른 단계 상승

0 초기값 변수 및 객체 설정

```
import java.util.Random;
import java.util.Scanner;

public class _1박2이 {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        Random ran = new Random();

        String[] level1 = {"ㄱ", "ㄴ", "ㄷ", "ㄹ", "ㅁ", "ㅂ", "ㅅ", "ㅇ", "ㅈ", "ㅊ", "ㅋ", "ㅌ",
            "표", "ㅎ", "ㅊ", "ㅊ", "ㅊ", "ㅊ", "ㅊ", "ㅊ", "ㅊ", "ㅊ", "ㅊ", "ㅊ", "ㅊ"};
        String[] level2 = {"가", "나", "다", "라", "마", "바", "사", "아", "자", "차", "카", "타", "파", "하",
            "몸", "삼", "꿈", "물", "불", "피", "잠", "새", "옷", "집", "훈", "글", "길", "눈",
            "점", "방", "빛", "코", "간", "잎", "새", "강", "양", "말", "산", "귀", "신", "빵"};
        String[] level3 = {"배우", "주스", "페지", "지하", "사수", "기체", "쥬스", "아구", "코드", "이찬",
            "자바", "자료", "마치", "하이", "동상", "게임", "점수", "원본", "시작", "코딩",
            "레벨", "문장", "이중", "차원", "정확", "수원", "휴먼", "교육", "센터", "객체"};
        String[] level4 = {"텀블러", "색종이", "컴퓨터", "선생님", "조건문", "반복문", "모니터", "마우스", "키보드",
            "비상구", "책가방", "구글링", "닉네임", "지우개", "깃허브", "검색기", "이미지", "동영상",
            "더보기", "박진우", "이원재", "쥬스바", "수박바", "아이스", "출입문", "한글날", "유튜브",
            "자료형", "형광등", "화이트", "그레이", "아파트", "상록수", "침엽수", "수원시", "인계동"};
        String[] level5 = {"고진감래", "구사일생", "과유불급", "괄목상대", "근목자흑", "다다익선", "대동소이", "동병상련",
            "문전성시", "백발백중", "사면초가", "살신성인", "설상가상", "아전인수", "역지사지", "우이독경",
            "자업자득", "죽마고우", "청출어람", "촌철살인", "타산지석", "토사구팽", "파죽지세", "풍전등화",
            "함흥차사", "동상이몽", "일박이이"};

        String[][] level = {level1, level2, level3, level4, level5};
```

- 입력을 위한 Scanner 객체
- 랜덤한 값을 생성하는 Random 객체
- 각 Level당 나올 단어를 저장한 level 배열

1 게임 시작/종료 및 단계 설정

```
while (true) {
    int levelCount = 0;
    int saveJumsu = 0;
    int jumpsu = 0;
    int totalJumsu = 0;
    boolean check = false;
    System.out.print("▶ 선택해주세요. [ 1 : 시작 / 0 : 종료 ] ");
    String startEnd = scan.nextLine();

    if (startEnd.equals("1")) {

    } else if (startEnd.equals("0")) {
        System.out.println("게임을 종료했습니다.");
        break;
    } else {
        System.out.println("※ 다시 선택해주세요. \n");
    }
}
scan.close();
```

➡ 프로그램이 실행되고 게임에 필요한 변수가 생성된 뒤 입력받은 값에 따라 게임 시작/종료가 되도록 한다.

```
if (startEnd.equals("1")) {
    while (true) {
        System.out.print("\n▶ 레벨을 선택해주세요 [ 1~5 : 난이도 / 0 : 처음 화면으로 ] ");
        String selectLevel = scan.nextLine();
        if (selectLevel.equals("1")) {
            levelCount = 0;
        } else if (selectLevel.equals("2")) {
            levelCount = 1;
        } else if (selectLevel.equals("3")) {
            levelCount = 2;
        } else if (selectLevel.equals("4")) {
            levelCount = 3;
        } else if (selectLevel.equals("5")) {
            levelCount = 4;
        } else if (selectLevel.equals("0")) {
            System.out.println("처음 화면으로 돌아갑니다.");
            levelCount = level.length;
        } else {
            System.out.println("다시 입력해주세요");
            continue;
        }
        break;
    }
}
```

➡ 게임이 시작되고 입력받은 값에 따라 게임 단계를 설정하고, 단계에 따른 게임을 실행한다.

2 2차원배열에 담긴 단어를 랜덤으로 사용

```
for (int i=levelCount; i<level.length; i++) {  
    if (jumsu != 0) {  
        jumsu = 0;  
        break;  
    }  
    float avgCount = 0;  
    for (int j=1; j<=15; j++) {  
        System.out.println("▶ " + (i+1) + "단계 (" + j + "/15)" + " ◀");  
        String levelText = level[i][ran.nextInt(level[i].length)];  
        System.out.println("단어 : " + levelText);  
        System.out.print("입력 : ");  
        String text = scan.nextLine();
```



단계 설정을 한 뒤 게임 단계를 지정하는 i for문과 각 단계별 게임이 진행되는 j for문이 실행된다.



게임 진행 시 각 단계(i)에 맞는 랜덤하게 생성된 값과 플레이어가 입력받은 입력값을 각 변수에 저장한다.

각 단계에 맞는 값은 2차원배열의 형태로 저장되어 있으므로, 랜덤한 값을 저장하기 위해 `level[i][ran.nextInt(level[i].length)]`를 사용하였다.

3 정답/오답에 따른 점수 및 정확도 계산

```
if (text.equals(levelText)) {  
    avgCount++;  
    jumsu += 20;  
    System.out.println("\n★ 정답입니다(점수+20)");  
} else {  
    jumsu -= 10;  
    System.out.println("\n◆ 틀렸습니다(점수-10)");  
}  
System.out.printf("★ 현재 점수 : " + jumsu + " / 정확도 : " + "%.1f" + "%\n\n", ((avgCount/j)*100));
```

➡ 점수 및 정확도를 계산하기 위해 랜덤하게 생성된 값과 플레이어가 입력받은 입력값을 비교하여 생성값과 입력값이 정답/오답일 경우의 조건을 걸어준다.

정답(True) : 정확도 계산을 위한 카운트와 점수를 증가(+20)하고, 정답이라는 출력문을 출력한다.

오답(False) : 점수를 감소(-10)시키고, 틀렸다는 출력문을 출력한다.

그리고 현재 시도 횟수(j)의 점수와 정확도를 보여준다. (정확도는 소수점 1자리까지 보여준다.)

4 점수에 따른 변화

```
while (j == 15) {  
    if (jumsu <= 30) {  
        System.out.println("◇ " + (i+1) + "단계 실패..");  
        System.out.printf("☆ 총 점수 : " + jumsu + " / 정확도 : " + "%.1f" + "%\n\n", ((avgCount/j)*100));
```



15번의 시도 끝에 점수가 30점 이하일 시,
단계 실패 문구 및 정확도와 점수가 출력

```
System.out.print("\n▶ 선택해주세요. [ 1 : 1단계부터 도전 / 2 : 처음 화면으로 ]");
```



또한 해당 구문을 출력 후 입력을 받음

```
String select = scan.nextLine();  
if (select.equals("1")) {  
    System.out.println("1단계부터 다시 도전합니다. \n");  
    jumsu = 0; |  
    i = -1;
```



1을 입력받은 경우 해당 단계를 다시 도전
점수를 = -1 로 설정하여 증감식이 적용될 때
현재 레벨로 돌아올 수 있게 됨

```
} else if (select.equals("2")) {  
    System.out.println("처음 화면으로 돌아갑니다. \n");  
    jumsu = 1;  
} else {  
    System.out.println("※ 다시 선택해주세요. \n");  
    continue;
```



2를 입력받은 경우 처음 화면으로 이동
점수가 0이 아니면 i반복문을 종료 가능하기에 jumsu 값을 1로 설정
1, 2가 아닌 값이 입력되면 다시 선택하도록 함

5 30점 초과, 100점 미만일 시

```
if (jumsu > 30 && jumsu < 100) {  
    System.out.println("◆ " + (i+1) + "단계 종료, 100점을 노려보세요!");  
    System.out.printf("☆ 총 점수 : " + jumsu + " / 정확도 : " + "%.1f" + "%\n\n", ((avgCount/j)*100));
```



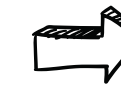
30점 초과, 100점 미만일 시

```
System.out.print("\n▶ 선택해주세요. [ 1 : 재도전 / 2 : 처음 화면으로 ] ");  
String select = scan.nextLine();
```



이와 같은 메뉴가 뜬 뒤 입력을 받는다

```
if (select.equals("1")) {  
    System.out.println((i+1) + "단계를 재도전합니다. \n");  
    jumsu = 0;  
    i -= 1;
```



1을 입력받은 경우 1단계부터 도전
점수를 0으로 설정하여 증감식이 적용
i=-1임으로 이후 i++ => i = 0이 되어 1단계로 회귀

```
} else if (select.equals("2")) {  
    System.out.println("처음 화면으로 돌아갑니다. \n");  
} else {  
    System.out.println("※ 다시 선택해주세요. \n");  
    continue;
```



2를 입력받은 경우 처음 화면으로 이동
점수가 0이 아니면 i반복문을 종료 가능하기에 jumsu 값을 1로 설정
1, 2가 아닌 값이 입력되면 다시 선택하도록 함

6 점수가 100 이상인 경우 - 최고 단계

```

if (jumsu >= 100) {
    totalJumsu += jumsu;
    if (i == level.length-1) {
        System.out.println("
        + CONGRATULATION!
        +
        +
        +
        System.out.println("★ 마지막 단계 클리어! / 누적 점수 : " + totalJumsu + " ★");
        System.out.println("▶ 마지막 단계를 클리어 하셨습니다. 어떻게 하시겠습니까?");
        while (true) {
            System.out.print("\n▶ 선택해주세요. [ 1 : 1단계부터 도전 / 2 : 재도전 / 3 : 처음 화면으로 ] ");
            ...
        }
    }
}

System.out.print("\n▶ 선택해주세요. [ 1 : 1단계부터 도전 / 2 : 재도전 / 3 : 처음 화면으로 ] ");
String select = scan.nextLine();
    
```

⇒ 문구 출력 후, 다음 선택지 제시

⇒ 입력받은 값(1~3)에 따라 조건 분리

1(1단계부터)을 입력받은 경우

```

if (select.equals("1")) {
    System.out.println("1단계부터 다시 도전합니다. \n");
    jumsu = 0;
    i = -1;
    break;
}
    
```

1단계부터 다시 시작 // i=-1;

```

    }
    break;
}
}
    
```

for(i)

7 점수가 100 이상인 경우 - 최고 단계

2(현재 재도전)를 입력받은 경우

```
} else if (select.equals("2")) {  
    System.out.println((i+1) + "단계를 재도전합니다. \n");  
    jumsu = 0;  
    i -= 1;  
    break;  
}
```

⇒ 현재 단계 재도전을 위해 i-=1;

for(i)
break;

3(처음 화면)을 입력받은 경우

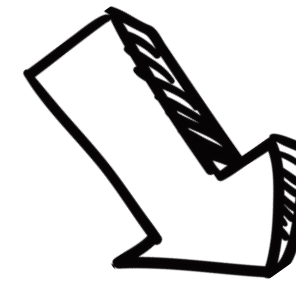
```
} else if (select.equals("3")) {  
    System.out.println("처음 화면으로 돌아갑니다. \n");  
    break;  
} else {  
    System.out.println("※ 다시 선택해주세요.");  
}
```

⇒ 그 외는 while문 반복

8 점수가 100 이상인 경우 - 최고 단계

```
for (int i=levelCount; i<level.length; i++) {  
    if (jumsu != 0) {  
        jumsu = 0;  
        break;  
    }  
    float avgCount = 0;  
    for (int j=1; j<=15; j++) {  
  
        ...  
        ...  
        ...  
    }  
    break; // 조건 선택이 끝난 경우 j반복문 종료  
}
```

점수가 존재하는 경우(=처음 화면으로 돌아가기)
- i 반복문 종료



```
// 게임 시작  
while (true) {  
    int levelCount = 0;  
    int saveJumsu = 0;  
    int jumsu = 0;  
    int totalJumsu = 0;  
    boolean check = false;  
    System.out.print("▶ 선택해주세요. [ 1 : 시작 / 0 : 종료 ] ");  
    String startEnd = scan.nextLine();
```

9 점수가 100 이상인 경우 - 최고 단계가 아닌 경우

```
System.out.println("★ " + (i+1) + "단계 클리어! ★");
System.out.printf("☆ 총 점수 : " + jumsu + " / 정확도 : " + "%.1f" + "%\n", ((avgCount/j)*100));
System.out.println("☆ 누적 점수 : " + totalJumsu);
while (true) {
    System.out.print("\n▶ 선택해주세요. [ 1 : 1단계부터 도전 / 2 : 재도전 / 3 : 다음 단계 / 4 : 처음 화면으로 ");
    if (saveJumsu >= 100 && (avgCount/j*100) == 100) {
        System.out.print("/ 5 : 최고 단계로 ] ");
        check = true;
    } else {
        System.out.print("] ");
    }
    String select = scan.nextLine();
}
```

⇒ 차이점

```
System.out.println("1단계부터 다시 도전합니다. \n");
totalJumsu = 0;
saveJumsu = 0;
```

```
System.out.println((i+1) + "단계를 재도전합니다. \n");
totalJumsu -= jumsu;
```

```
System.out.println("다음 단계로 넘어갑니다. \n");
saveJumsu = jumsu;
```

10 점수가 100 이상인 경우 - 최고 단계가 아닌 경우 - 숙련자

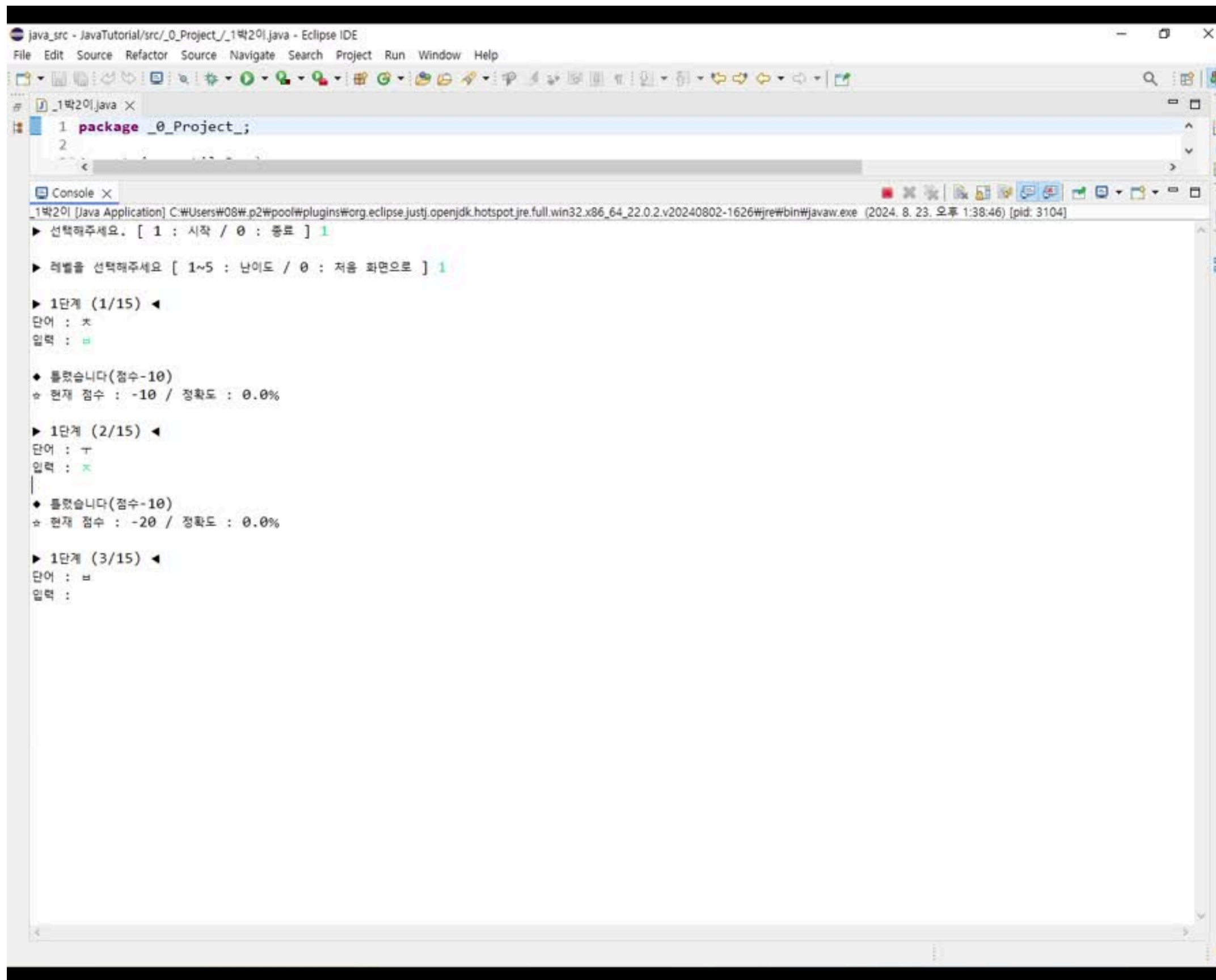
```
boolean check = false;
```

```
while (true) {  
    System.out.print("\n▶ 선택해주세요. [ 1 : 1단계부터 도전 / 2 : 재도전 / 3 : 다음 단계 / 4 : 처음 화면으로 ");  
    if (saveJumsu >= 100 && (avgCount/j*100) == 100) {  
        System.out.print("/ 5 : 최고 단계로 ");  
        check = true;    이전 단계 점수가 100점 이상, 현재 단계 정확도가 100%인 경우  
    } else {  
        System.out.print("] ");  
    }  
      
    } else if (select.equals("5") && check == true) {  
        jumpsu = 0;  
        i = 3;  
        break;  
    }
```

증감식 적용 고려 ➡ i=3;

check == true를 이용하여 “최고 단계로” 가 표현될 때만 누르기 가능

실행 화면



```
java_src - JavaTutorial/src/_0_Project/_1박2이.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help

_1박2이.java
1 package _0_Project_;
2

Console
_1박2이 [Java Application] C:\Users#08#.p2#pool#plugins#org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-1626#jre#bin#javaw.exe (2024. 8. 23. 오후 1:38:46) [pid: 3104]

▶ 선택해주세요. [ 1 : 시작 / 0 : 종료 ] 1

▶ 레벨을 선택해주세요 [ 1~5 : 난이도 / 0 : 처음 화면으로 ] 1

▶ 1단계 (1/15) ◀
단어 : ㅈ
입력 : ㅈ

◆ 통했습니다(점수-10)
* 현재 점수 : -10 / 정확도 : 0.0%

▶ 1단계 (2/15) ◀
단어 : ㅊ
입력 : ㅊ

◆ 통했습니다(점수-10)
* 현재 점수 : -20 / 정확도 : 0.0%

▶ 1단계 (3/15) ◀
단어 : ㅂ
입력 :
```

1박2이 : 타자게임 프로젝트

바다 201

감사합니다.

이찬 | 박진우 | 이원재