

# Java mini project

휴먼교육센터- 우승훈, 전민정

01

주제 소개

02

기술 기능 선정 회의 결과

03

개인별 테스트 코드 설명

04

최종 리팩토링 코드- 선정기준/완성본

05

구현 영상





## 타자게임프로그램

단어가 자동으로 콘솔 창에 나오면 지시 단어를 사용  
자 타이핑 하고, 그 결과를 채점하는 프로그램

선정 이유: 타자 게임은 문자열의 정확성을 통해 채점을 하므로 조건문과 반복문, `.charAt()`메소드 기능으로 구현할 수 있다. 따라서 지금까지 배운 문법으로 타자게임이 구현 시키기에 적합하다고 판단하여 선정하게 되었다.

### 게임의 과정

1. 사용자는 아이디를 입력한다.
2. 저장된 단어를 랜덤으로 나오게 한다. 단어는 중복되어 나오지 않는다.
3. 사용자는 제시된 단어를 입력한다.
4. 제시된 단어(띄어쓰기, 대소문자)등을 고려하여 정확히 입력하면 정답, 아니면 오답이다.
5. 저장된 단어가 다 소진되면 게임 한판이 종료된다.
6. 사용자의 점수를 저장한다.
7. 사용자는 자신의 점수와 랭킹을 확인할 수 있다.
8. 사용자는 재도전과 게임 종료를 선택할 수 있다.





아이디 입력	<ul style="list-style-type: none"> <li>Scanner 클래스로 사용자의 아이디를 입력 받는다. 이를 배열에 저장한다.</li> </ul>
중복 없이 단어 랜덤 추출	<ul style="list-style-type: none"> <li>배열에 이미 저장된 단어를 랜덤으로 추출하고 중복되지 않게 새 배열에 저장하여 타자게임 시 새 배열에서 단어를 가져온다.</li> </ul>
채점하기	<ul style="list-style-type: none"> <li>사용자가 입력한 문자열과 제시된 문자열을 .charAt() 메소드로 비교한 뒤 정확히 일치하면 정답, 아닐 시 오답이라고 출력한다.</li> <li>연속 정답일 경우 보너스점수를 5점을 누적하여 받는다.</li> <li>점수를 배열에 저장한다.</li> </ul>
점수 출력	<ul style="list-style-type: none"> <li>정답 개수와 오답 개수를 세어 정답률(%)과 보너스점수를 출력한다.</li> <li>최종 점수는 정답 개수/문제 개수*100 + 보너스점수이다.</li> <li>점수는 100점을 넘어갈 수 있다.</li> </ul>
랭킹 출력	<ul style="list-style-type: none"> <li>랭킹을 내림차순으로 정렬한다.</li> <li>점수 내림차순에 따라 아이디와 랭킹이 각각 배열의 동일한 인덱스 번호에 저장될 수 있도록 한다.</li> </ul>



```
10
11
12     Scanner in = new Scanner(System.in);
13
14     String[] idbx = new String[100];
15     double[] rank = new double[100];
16
17     System.out.println("id를 입력하세요");
18     String id=in.nextLine();
19     for(int i = 0; i < idbx.length; i++) {
20         if (idbx[i] == null) {
21             idbx[i] = id;
22             break;
23         }
24     }
25     System.out.println "["+id+"] "+id가 입력되었습니다");
26
27
28
```

배열의 값이 null일 경우 id를 저장한다. 저장 후, 과정 중단

Problems Javadoc Declaration Console Call Hierarchy

<terminated> H004 [Java Application] C:\Program Files\Java\jre1.8.0\_261\bin\javaw.exe (2024. 3. 8. 오후 3:30:00)

id를 입력하세요

d

[d]id가 입력되었습니다

Scanner 클래스를 이용하여 아이디를 입력받았고, 아이디를 변수 idbx라는 배열에 id를 저장한다.



랜덤배열 확인(game 배열)

```

29
30
31 Random r = new Random();
32 String word[] = {"안녕", "구름은", "야자나무", "사과", "바나나", "배"};
33 String word2[] = {"안녕", "구름은", "야자나무", "사과", "바나나", "배"};
34 String []game= new String[word.length];
35
36 int d=-1; //game 배열의 인덱스
37
38 System.out.println("로딩중입니다.....");
39 for(int i=0; i<word.length; i++) {
40     int c=r.nextInt(word.length); //다시 뽑기
41     for(int j=0; j<word2.length; j++) {
42         if(word[c]==word2[j]){ //다 비교해보고 같으면
43             d++;
44             game[d]=word[c];
45             System.out.println(game[d]);
46             word2[j]=""; //word2에 공백을 집어넣기
47         } else {
48             i--; //다시뽑기
49         }
50     }
51 }
52

```

전부 비교해보고 해당 단어가 없으면  
다시 뽑는다.

로딩중입니다.....

배  
구름은  
야자나무  
사과  
바나나  
안녕

로딩중입니다.....

사과  
배  
바나나  
안녕  
구름은  
야자나무

변수 word 배열에 미리 단어를 저장한다. 변수 word2 라는 배열 하나를 더 만들어 똑같은 단어를 저장한다. 중복없이 랜덤으로 단어를 추출하기 위해 변수 word 배열에서 출력된 값과 변수 word2 배열의 값과 비교한다. 한 번 출력하면, word2의 똑같은 단어는 공백으로 저장된다. 이 과정을 반복하여 랜덤 추출된 단어를 변수 game이라는 배열에 재저장하여 타자 게임 시 game배열에서 하나씩 가져오도록 하였다.



```

52
53     System.out.println("게임을 시작합니다. 제시된 단어를 똑같이 입력하세요");
54
55     int correct=0;
56     int wrong=0;
57     int cnt=0;
58     int bonus=0;
59     int sum=0;
60
61     for(int i=0;i<game.length; i++){
62
63         int b=0;
64
65         System.out.println(game[i]);
66         String userword=in.nextLine();
67
68         if (game[i].length() == userword.length()) {
69             for(int k=0; k<userword.length(); k++) {
70                 if(userword.charAt(k)==game[i].charAt(k)) {
71                     b++;
72                 }
73             }
74
75             if (b == userword.length()) {
76                 System.out.println("정답입니다");
77                 correct++;
78                 cnt++;
79
80                 if(cnt>=2) {
81                     bonus+=5;
82                     System.out.println("보너스 점수는 누적 +" +bonus+"입니다.");
83                     sum+=5;
84                 }
85             }
86
87         }else {
88             System.out.println("오답입니다");
89             wrong++;
90             cnt=0;

```

```

91             bonus=0;
92         }
93
94         }else {
95             System.out.println("오답입니다");
96             wrong++;
97             cnt=0;
98             bonus=0;
99         }
100 //

```

사용자가 입력한 단어와 제시어의 길이가 같으면 비교를 시작한다. 두 단어를 비교하여 정확히 일치하면 정답,아니면 오답이다.

연속 정답일 경우 보너스 점수 5점을 누적하여 받는다. 오답일 경우 보너스 점수가 초기화된다.

정답 개수와 오답 개수를 센다.



정답입니다

배

배

정답입니다

보너스 점수는 누적 +5입니다.

안녕

안녕

오답입니다

구름은

구름은

정답입니다

<

<terminated> H004 [Java Application] C:\Program Files\Java\jre

구름은

정답입니다

야자나무

야자나무

정답입니다

보너스 점수는 누적 +5입니다.

바나나

바나나

정답입니다

보너스 점수는 누적 +10입니다.



```

}double per = ((double)correct/word.length)*100;
double score =per+sum;
System.out.println("-----");
System.out.println("당신은"+word.length+"개 중"+ correct+"개 정답이고"+wrong+"개 오답임");
System.out.println("정답률은"+per+"%입니다.");
System.out.println("최종 보너스 점수는"+sum+"입니다.");
System.out.println("최종 점수는"+score+"점 입니다.");
System.out.println("-----");//정산 시작

```

```

당신은6개 중2개 정답이고4개 오답임
정답률은33.333333333333333333%입니다.
최종 보너스 점수는5입니다.
최종 점수는38.333333333333333333점 입니다.
-----

```

최종 점수가 출력된다. 최종 점수는 정답 개수/문제 개수\*100(정답률) 과 보너스 점수가 추가된 점수이다.



사용자의 점수를 변수 rank배열에 저장하고 내림차순 정렬을 한다.  
 이때, 사용자의 아이디와 점수의 인덱스 번호를 일치시키기위해서 점수가 내림차순 정렬이 되면 동일하게 아이디도 정렬시키는 방식으로 하였다.

변수 Rank와 idbx의 길이는 100이므로 null값이 아닌 것만 출력하게 하였다.

게임이 종료된다.

```
for(int i = 0; i < rank.length; i++) {
    if (rank[i] == 0) {
        rank[i] = score;
        break;
    }
}
```

rank 배열의 값이 null일 경우 score를 저장한다. 저장 후, 과정 중단

```
System.out.println("RANK");
System.out.println("★★★★★★★★★★★★★★★★★★★★");
for(int i=0; i<idbx.length; i++) {
    for(int j=0; j<i; j++) {
        if(rank[i]>rank[j]) {
            double temp=rank[i];
            rank[i]=rank[j];
            rank[j]=temp; //점수 내림차순
            String temp2=idbx[i];
            idbx[i]=idbx[j];
            idbx[j]=temp2; //점수와 같이 id 내림차순
        }
    }
}

for(int i=0; i<idbx.length; i++) {
    if(idbx[i]!=null) {
        System.out.println(idbx[i]+":"+rank[i]+"점");
    }
}
System.out.println("★★★★★★★★★★★★★★★★★★★★");
System.out.println("게임 종료, retry? 새로고침");
```

```
RANK
★★★★★★★★★★★★★★★★★★★★
○○○:38.33333333333333점
★★★★★★★★★★★★★★★★★★★★
게임 종료, retry? 새로고침
```



# 테스트 코드 시연

전민정

```
id를 입력하세요
java131
[java131]id가 입력되었습니다
로딩중입니다.....
게임을 시작합니다. 제시된 단어를 똑같이 입력하세요
사과
사과
정답입니다
-----
바나나
바나나
정답입니다
-----
보너스 점수는 누적 +5입니다.
-----
안녕
안녕
정답입니다
-----
보너스 점수는 누적 +10입니다.
-----
구름은
구름
오답입니다
-----
야자나무
야자나무
정답입니다
-----
배
배
정답입니다
-----
보너스 점수는 누적 +5입니다.
-----
당신은6개 중5개 정답이고1개 오답임
```

```
-----
당신은6개 중5개 정답이고1개 오답임
정답률은83.33333333333334%입니다.
최종 보너스 점수는15입니다.
최종 점수는98.33333333333334점 입니다.
-----
```

RANK

```
★★★★★★★★★★★★★★★★★★★★
java131:98.33333333333334점
★★★★★★★★★★★★★★★★★★★★
게임 종료, retry? 새로고침
```



```
Scanner in = new Scanner(System.in);
Random r = new Random();
String[] id = new String[100];
double[] idScore = new double[id.length];
System.out.println("타자 게임을 시작합니다.");
for(;;) {
    System.out.println("id를 입력하세요.");
    String idCr = in.nextLine();
    System.out.println("[ "+idCr+" ] "+"입력되었습니다."); //id 입력
```

id를 저장하기 위해 길이가 100인 정수배열의 변수 id 선언하고 최종 점수를 저장하기 위해 길이가 변수 id와 같은 실수배열의 변수 idScore 선언한다.  
게임 시작 문구 출력 후 id를 자료형 String인 변수 idCr에 입력하여 저장한다.



```
String[] wordKr = {"사과", "바나나", "배", "감", "귤", "대추", "포도", "샤인머스켓", "딸기", "자몽", "레몬", "라임"};
String[] sentenceEn = {"The used key is always bright.", "The best things in life are free."
    , "A barking dog never bites.", "No pain No gain."
    , "A good medicine tastes bitter.", "It never rains but it pours."
    , "Walls have ears.", "HELLO WORLD!", "Do not judge a book by its cover."
    , "A drowning man will catch at a straw."
    , "Between a rock and a hard place.", "Ewwwww!"}; //레벨 별 출력되는 문자열
String[] game = new String[12]; //게임에서 사용되는 문자열 저장 배열
```

게임에서 입력해야하는 문자열을 미리 지정하여 자료형 String배열의 변수 wordKr과 sentenceEn에 각각 대입한다.

그리고 출력되는 문자열의 중복을 방지하기 위해 게임에 출력되는 문자열을 저장하도록 길이가 12인 자료형 String배열의 변수 game을 선언한다.



우승훈

```
String level;
for(;;) {
    System.out.println("레벨을 선택하세요.");
    System.out.println("1(한글단어) / 2(영어문장)");
    System.out.print("레벨선택 >> ");
    level = in.nextLine();
    if (level.charAt(0) == '1') {
        System.out.println("레벨 1(한글단어)을 선택했습니다.");
        System.out.println("-----");
        for(int i = 0; i < wordKr.length; i++) {
            game[i] = wordKr[i];
        }
    } else if (level.charAt(0) == '2') {
        System.out.println("레벨 2(영어문장)를 선택했습니다.");
        System.out.println("-----");
        for(int i = 0; i < sentenceEn.length; i++) {
            game[i] = sentenceEn[i];
        }
    } else {
        System.out.println("잘못 입력하셨습니다. 레벨은 1 또는 2 입니다.");
        continue;
    }
    break;
} //레벨 선택
```

게임에서 출력되는 문자열을 사용자가 선택하도록 한다.  
사용자가 1 또는 2만 입력하도록 유도하고 입력한 레벨에 따라 변수 wordKr 또는 sentenceEn의 값을 변수 game에 차례로 저장한다.



```
String[] printWord = new String[10]; //배열 길이: 제시되는 문자열의 수
String[] exWord = new String[printWord.length]; //중복 체크 위한 배열
printWord[0] = game[r.nextInt(game.length)]; //확정된 문자열 저장의 배열
exWord[0] = printWord[0]; //첫번째 문자열을 미리 저장
for (int i = 1; i < printWord.length; i++) { //i는 printWord의 인덱스 번호
    printWord[i] = game[r.nextInt(game.length)]; //game에서 임의의 문자열을 배열에 저장
    for (int j = 0; j < i; j++) {
        if (printWord[i] == exWord[j]) {
            i--;
            break;
        } else {
            if (j == i-1) {
                exWord[i] = printWord[i];
            }
        }
    }
}
//중복 제거 저장
```

게임에서 출력되는 문자열의 중복을 제거하기 위해 자료형이 String배열인 변수 printWord와 exWord를 각각 선언한다.

exWord에는 확정된 문자열들을 저장하여 printWord에 대입되는 랜덤한 문자열의 중복 여부를 비교할 수 있도록 한다.



```
String user; //사용자가 키보드로 입력하는 값 저장
int pass = 0; // 정답 갯수
int block = 0; // 오답 갯수
double tfPer = 0; //정답률
int passSeq = 0; //2개 이상 연속으로 정답을 맞춘 횟수
double bonus = 0; //보너스 점수
int passCnt = 0; // 정답 여부 판별
System.out.println("게임을 시작합니다. 제시되는 글을 똑같이 입력하세요.");
System.out.println("정확히 입력할 경우 점수가 오릅니다.");
System.out.println("-----");
```

게임을 진행하는 데 필요한 값들을 저장할 변수를 선언한다.  
게임이 시작된다는 문자열과 게임 플레이 방법을 알려주는 문자열을 출력한다.



사용자가 콘솔에 출력되는 문자열을 따라서  
입력하는 값에 따라 정답 및 오답의 여부를 판단한다.  
콘솔에 출력된 문자열과 사용자가 입력한 문자열의  
문자들을 비교하면서 같을 경우 정답, 아닐 경우  
오답으로 판단한다.

2번 이상 연속으로 정답일 경우 보너스 점수를  
더하고, 오답이 나올 경우 보너스 점수를 초기화한다.

```
for(int i = 0; i < printWord.length; i++) {
    System.out.println("제시어: "+printWord[i]);
    System.out.print("입력 >> ");
    user = in.nextLine();
    if (user.length() == printWord[i].length()) {
        passCnt = 0;
        for(int j = 0; j < user.length(); j++) {
            if (user.charAt(j) == printWord[i].charAt(j)) {
                passCnt++;
            }
        }
        if (passCnt == printWord[i].length()) {
            System.out.println("정답입니다.");
            System.out.println("-----");
            pass++;
            passSeq++;
            if (passSeq >= 2) {
                bonus+=5;
            }
        } else {
            System.out.println("오답입니다.");
            System.out.println("-----");
            block++;
            passSeq = 0;
            bonus = 0;
        }
    } else {
        System.out.println("오답입니다.");
        System.out.println("-----");
        block++;
        passSeq = 0;
        bonus = 0;
    }
}
```



```
for(int i = 0; i < printWord.length; i++) {  
    System.out.println("제시어: "+printWord[i]);  
    System.out.print("입력 >> ");  
    user = in.nextLine();  
    if (user.equals(printWord[i])) {  
        System.out.println("정답입니다.");  
        System.out.println("-----");  
        pass++;  
        passSeq++;  
        if (passSeq >= 2) {  
            bonus+=5;  
        }  
    } else {  
        System.out.println("오답입니다.");  
        System.out.println("-----");  
        block++;  
        passSeq = 0;  
        bonus = 0;  
    }  
}
```

.equals() 메서드를 활용하여 정답 및 오답 여부를 판단한 코드



```
tfPer = ((double)pass/printWord.length)*100;    //정답률
double score = tfPer+bonus;    //최종 점수
System.out.println("라운드가 종료되었습니다.");
System.out.println("정답은 "+pass+"개이고, "+ "오답은 "+block+"개 입니다.");
System.out.print("정답률은 "+tfPer+"%이고, ");
System.out.println("보너스 점수는 "+bonus+"점 입니다.");
System.out.print("유저 "+idCr+"의 ");
System.out.println("최종 점수는 "+(score)+"입니다.");
System.out.println("-----");    //게임 진행 및 점수 정산
```

라운드 종료 후 점수를 산정한다.

정답을 입력한 횟수를 저장해 정답률을 출력했고 보너스 점수를 그 수에 더해 최종 점수를 출력한다.



```
for(int i = 0; i < id.length; i++) {  
    if (id[i] == null) {  
        id[i] = idCr;  
        idScore[i] = score;  
        break;  
    }  
} //유저 id와 산정된 score를 랭크 출력 위해 배열에 저장
```

입력한 id와 산정된 최종 점수를 변수 id와 idScore의 동일한 인덱스 번호에 저장하여 사용자의 id와 최종 점수를 랭킹에 반영할 수 있도록 한다.



```
double exScr;
String exId;
for(int i = 0; i < id.length; i++) {
    for(int j = 0; j < i; j++) {
        if (idScore[i] >= idScore[j]) {
            exScr = idScore[i];
            idScore[i] = idScore[j];
            idScore[j] = exScr;
            exId = id[i];
            id[i] = id[j];
            id[j] = exId;
        }
    }
} //처음에 입력한 id와 게임을 플레이한 후 산정된 score값을 배열에 넣고 인덱스번호가 작은 것부터 내림차순으로 동시에 정렬하는 코드
```

변수 id와 idScore에 저장한 값을 변수 idScore의 값에 대한 선택 정렬을 통해 내림차순으로 정렬하고, 변수 id의 값을 같이 정렬한다.



```
for (;;) {
    System.out.println("랭킹을 확인하시겠습니까?");
    System.out.println("1(예) / 2(아니오)");
    System.out.print("랭킹확인 >> ");
    String rankChk;
    rankChk = in.nextLine();
    if (rankChk.charAt(0) == '1') {
        System.out.println("-----");
        for(int i = 0; i < id.length; i++) {
            if (id[i] != null) {
                System.out.println((i+1)+"위 "+"["+id[i]+" "+idScore[i]+"점 ");
            }
        }
    } else if (rankChk.charAt(0) == '2') {
    } else {
        System.out.println("잘못 입력했습니다. 1을 눌러 확인하거나, 2를 눌러 확인하지 않습니다.");
        continue;
    }
    break;
} //랭킹 확인
```

라운드 종료 후 사용자가 랭킹 확인 여부를 입력하도록 유도한다.  
선택 정렬을 통해 내림차순으로 정렬한 변수 id와 idScore값을 출력한다.  
이때 변수 id의 인덱스 번호 i의 값이 null값일 경우 출력하지 않는다.



```
System.out.println("-----");
System.out.println("한 게임이 종료되었습니다.");
System.out.println("게임을 재도전하시겠습니까?");
System.out.print("1(예) / ");
System.out.println("아무거나 누르기(아니오)(저장된 id와 score가 초기화됩니다.)");
String retry = in.nextLine();
if (retry.charAt(0) == '1') {
    System.out.println("게임을 재도전합니다.");
    continue;
} else {
    System.out.println("게임을 종료합니다.");
    break;
} //게임 재도전 여부
```

게임이 종료되었을 시 게임 재도전 여부 확인을 위한 코드



# 테스트 코드 시연

우승훈

```
타자 게임을 시작합니다.  
id를 입력하세요.  
|
```





- 아이디를 입력받고 배열에 저장시킬 때 처음에 입력받고 배열에 바로 저장하는 방법과 게임 라운드가 종료되고 저장하는 방법으로 구성시킨게 서로 달랐는데 라운드가 종료된 후 랭킹을 출력한다는 점에 있어서 최종 점수와 아이디를 라운드가 종료된 후 배열에 저장시키도록 최종 구성하였다.
- 전체적으로 두 사람의 코드가 유사하였지만 게임에 출력되는 문자열을 중복 제거하는 코드는 서로 달랐는데, 이때 '전민정'의 코드는 콘솔창에 중복 제거된 단어가 출력은 되지만 실행 속도가 눈에 띄게 느린 오류가 있었기에 최종 '우승훈'의 코드를 선정하였다.
- 입력받은 문자열을 채점하고 라운드 종료 후 점수를 산정하는 코드는 더 간결한 코드인 '전민정'의 코드를 선정하였다.
- 랭킹을 출력하는 코드는 두 코드를 재조합 하였다.
- 게임의 완성을 고려하여 게임의 level선택, 랭킹확인 여부 입력, 게임 재도전여부를 입력할 수 있는 '우승훈'의 코드를 추가하여 최종 구성하였다.





```
String[] id = new String[100];
double[] idScore = new double[id.length];
System.out.println("타자 게임을 시작합니다.");
    for(;;) {
        Scanner in = new Scanner(System.in);
        System.out.println("id를 입력하세요.");
        String idCr = in.nextLine();
        System.out.println("[ "+idCr+" ] "+"입력되었습니다.");
```

```
String[] wordKr = {"사과","바나나","배","감","귤","대추","포도","샤인머스켓","딸기","자몽","레몬","라임"};
String[] sentenceEn = {"The used key is always bright.", "The best things in life are free."
, "A barking dog never bites.", "No pain No gain.", "A good medicine tastes bitter."
, "It never rains but it pours.", "Walls have ears.", "HELLO WORLD!", "Do not judge a book by its cover."
, "A drowning man will catch at a straw.", "Between a rock and a hard place.", "Ewwwww!"};
String[] game = new String[12];
```



```

String level;
for(;;) {
    System.out.println("레벨을 선택하세요.");
    System.out.println("1(한글단어) / 2(영어문장)");
    System.out.print("레벨선택 >> ");
    level = in.nextLine();
    if (level.charAt(0) == '1') {
        System.out.println("레벨 1(한글단어)을 선택했습니다.");
        System.out.println("-----");
        for(int i = 0; i < wordKr.length; i++) {
            game[i] = wordKr[i];
        }
    } else if (level.charAt(0) == '2') {
        System.out.println("레벨 2(영어문장)를 선택했습니다.");
        System.out.println("-----");
        for(int i = 0; i < sentenceEn.length; i++) {
            game[i] = sentenceEn[i];
        }
    } else {
        System.out.println("잘못 입력했습니다. 레벨은 1~2 중 하나입니다.");
        continue;
    }
    break;
}

```



```
Random r = new Random();
String[] printWord = new String[5];
String[] exWord = new String[printWord.length];
printWord[0] = game[r.nextInt(game.length)];
exWord[0] = printWord[0];
for (int i = 1; i < printWord.length; i++) {
    printWord[i] = game[r.nextInt(game.length)];
    for (int j = 0; j < i; j++) {
        if (printWord[i] == exWord[j]) {
            i--;
            break;
        } else {
            if (j == i-1) {
                exWord[i] = printWord[i];
            }
        }
    }
}
```



```

int correct=0;
int wrong=0;
int cnt=0;
int bonus=0;
int sum=0;

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

int b=0;

System.out.println(printWord[i]);
String userword=in.nextLine();

        if (printWord[i].length() == userword.length()) {
            for(int k=0; k<userword.length(); k++) {
                if(userword.charAt(k)==printWord[i].charAt(k)) {
                    b++;
                }
            }

            if (b == userword.length()) {
                System.out.println("정답입니다");
                System.out.println("-----");
                correct++;
                cnt++;

                if(cnt>=2) {
                    bonus+=5;
                    System.out.println("보너스 점수는 누적 +" +bonus+"입니다.");
                    System.out.println("-----");
                    sum+=5;
                }

            }
        }else {

            System.out.println("오답입니다");
            System.out.println("-----");
            wrong++;
            cnt=0;
            bonus=0;

        }

    }else {

        System.out.println("오답입니다");
        System.out.println("-----");
        wrong++;
        cnt=0;
        bonus=0;

    }

}

```



```
double per = ((double)correct/printWord.length)*100;
double score =per+sum;
System.out.println("-----");
System.out.println("당신은"+printWord.length+"개 중"+ correct+"개 정답이고
                    "+wrong+"개 오답임");
System.out.println("정답률은"+per+"%입니다.");
System.out.println("최종 보너스 점수는"+sum+"입니다.");
System.out.println("최종 점수는"+score+"점 입니다.");
System.out.println("-----");
```

```
for(int i = 0; i < id.length; i++) {
    if (id[i] == null) {
        id[i] = idCr;
        idScore[i] = score;
        break;
    }
}
```



```
double exScr;
String exId;
for(int i = 0; i < id.length; i++) {
    for(int j = 0; j < i; j++) {
        if (idScore[i] >= idScore[j]) {
            exScr = idScore[i];
            idScore[i] = idScore[j];
            idScore[j] = exScr;
            exId = id[i];
            id[i] = id[j];
            id[j] = exId;
        }
    }
}
```



습니다.");



```
System.out.println("-----");
System.out.println("한 게임이 종료되었습니다.");
System.out.println("게임을 재도전하시겠습니까?");
System.out.print("1(예) / ");
System.out.println("2(아니오)(저장된 id와 score가 초기화됩니다.)");
String retry = in.nextLine();
if (retry.charAt(0) == '1') {
    System.out.println("게임을 재도전합니다.");
    continue;
} else {
    System.out.println("게임을 종료합니다.");
    break;
}
}
```



# 05

## 구현 영상

타자 게임을 시작합니다.  
id를 입력하세요.

I



★감사합니다★