

자바 프로그래밍 첫 미니 프로젝트

타자 게임

2024.08.23

정지용



주요 코드 소개

- 1 특수문자를 중복없이 배열에 저장하는 코드
- 2 게임에 사용될 문자열을 무작위로 섞는 코드
- 3 타자 속도를 구하기 위한 시간을 계산하는 코드
- 4 정확도를 구하기 위한 틀린 글자를 계산하는 코드
- 5 총점, 점수, 타자속도, 정확도의 값을 산출하는 코드
- 6 총점을 기준으로 순위를 정하는 코드

기능별 코드 설명

-게임에 사용될 문자열을 무작위로 섞기

```
int k = 0; k < gameTxt.length(); k++) {  
    char r_txt = gameTxt.charAt(r.nextInt(gameTxt.length())); // 게임 문자열에서 랜덤으로 하나 추출  
    // 밑에 반복문 시작을 위해 첫번째 값을 저장  
    if(randomGameTxt.length() == 0) {  
        randomGameTxt += r_txt;  
        System.out.println("첫 단어 : " + randomGameTxt);  
    }  
    // 인덱스 0 부터 저장된 게임 문자열의 길이까지 반복문 돌리기  
    for(int l = 0; l < randomGameTxt.length(); l++) {  
        System.out.println("문자 : " + r_txt);  
        if(randomGameTxt.charAt(l) != r_txt) {  
            System.out.println(randomGameTxt.charAt(l) + " 문자가 다름 " + r_txt);  
        }else {  
            if(k != 0) {  
                System.out.println("같은 랜덤 수 : " + r_txt);  
                System.out.println(randomGameTxt);  
                k--;  
                break;  
            }  
            break;  
        }  
    }  
    // 게임 문자열의 마지막 까지 중복 검사가 끝나면 추가  
    if(l == randomGameTxt.length() - 1) {  
        randomGameTxt += r_txt;  
        System.out.println(randomGameTxt);  
    }  
    // 게임 문자열 안에 섞어서 다 들어갔는지 길이를 비교해 확인  
    if(randomGameTxt.length() == gameTxt.length()) {  
        break;  
    }  
}
```

ex)

눈물!#\$ -> #물!눈\$



point

중복없이 모든 문자가
출력되어야 하기 때문에
앞에 첫 글자부터
저장된 모든 글자까지
중복검사를 진행

기능별 코드 설명

-점수 계산 및 오타 개수 구하기

```
if(userTxt.equals(randomGameTxt)) {  
    score += 20;  
}else {  
    missWord = "";  
    score += 0;  
}  
// 20240822_정지용_오타 개수 카운트  
for(int k = 0; k < randomGameTxt.length(); k++) {  
    if(userTxt.charAt(k) != randomGameTxt.charAt(k)) {  
        missCnt++;  
        missWord += userTxt.charAt(k) + " ";  
    }  
}
```



모든 글자가 맞아야만
점수를 부여
한 문제당 20점으로
100점 기준 5문제 제출

반복문을 통해
개별적으로 문자를 하나씩 비교
틀린 글자들은 화면에 출력하고
오타는 추후에 정확도 계산을 위해
missCnt를 증감

기능별 코드 설명

-총점, 점수, 타자속도, 정확도 산출하기

점수

```
if(userTxt.equals(randomGameTxt)) {  
    score += 20;  
}else {  
    missWord = "";  
    score += 0;  
}
```

타자속도

```
long beforeTime = System.currentTimeMillis();  
String userTxt = in.nextLine(); // 값을 입력받을 변수  
long afterTime = System.currentTimeMillis();  
long secDiffTime = (afterTime - beforeTime)/1000;
```

```
questionCnt++;  
accSpeed += secDiffTime;
```

```
userSpeed[i] = Math.round((accSpeed / questionCnt) * 10) / 10.0; // 소수점 1자리까지 끊어쓰기
```

정확도

```
for(int k = 0; k < randomGameTxt.length(); k++) {  
    if(userTxt.charAt(k) != randomGameTxt.charAt(k)) {  
        missCnt++;  
        missWord += userTxt.charAt(k) + " ";  
    }  
}
```

```
accTxt += userTxt;
```

```
userCorrect[i] = Math.round(((1 - (double) missCnt / accTxt.length()) * 100) * 10) / 10.0;
```

총점

```
for(int i = 0; i < userCnt; i++) {  
    userTotalScore[i] = Math.round(((userPoint[i] * 0.5) + userSpeedScore[i] + userCorrectScore[i]) * 100) / 100.0;  
}
```

점수

모든 글자가 맞으면 점수를 부여

타자 속도

입력값을 받기 전과 받은 후의 시간을 저장해서
두 값의 차이를 통해 입력 시간을 저장함
(누적시간 / 문제개수)를 통해
타자 속도를 소수점 1자리까지 산출

정확도

글자가 다르면 틀린 글자 변수를 증감시키고
사용자가 입력한 문자열을 누적해서 저장
(틀린 글자 / 전체 입력한 글자)를 통해
정확도를 소수점 1자리까지 산출

총점

100점 만점으로 점수 50%, 타자속도 20%,
정확도 30%를 기준으로 총점 산출



기능별 코드 설명

-게임에 사용될 문자열을 무작위로 섞기

```
int k = 0; k < gameTxt.length(); k++) {  
    char r_txt = gameTxt.charAt(r.nextInt(gameTxt.length())); // 게임 문자열에서 랜덤으로 하나 추출  
    // 밑에 반복문 시작을 위해 첫번째 값을 저장  
    if(randomGameTxt.length() == 0) {  
        randomGameTxt += r_txt;  
        System.out.println("첫 단어 : " + randomGameTxt);  
    }  
    // 인덱스 0 부터 저장된 게임 문자열의 길이까지 반복문 돌리기  
    for(int l = 0; l < randomGameTxt.length(); l++) {  
        System.out.println("문자 : " + r_txt);  
        if(randomGameTxt.charAt(l) != r_txt) {  
            System.out.println(randomGameTxt.charAt(l) + " 문자가 다름 " + r_txt);  
        }else {  
            if(k != 0) {  
                System.out.println("같은 랜덤 수 : " + r_txt);  
                System.out.println(randomGameTxt);  
                k--;  
                break;  
            }  
            break;  
        }  
    }  
    // 게임 문자열의 마지막 까지 중복 검사가 끝나면 추가  
    if(l == randomGameTxt.length() - 1) {  
        randomGameTxt += r_txt;  
        System.out.println(randomGameTxt);  
    }  
    // 게임 문자열 안에 섞어서 다 들어갔는지 길이를 비교해 확인  
    if(randomGameTxt.length() == gameTxt.length()) {  
        break;  
    }  
}
```

ex)

눈물!#\$ -> #물!눈\$



point

중복없이 모든 문자가
출력되어야 하기 때문에
앞에 첫 글자부터
저장된 모든 글자까지
중복검사를 진행

기능별 코드 설명

-총점을 기준으로 순위 정하기

```
int [] ranking = new int[userCnt];
double max_num = 0;
int max_index = 0;
int x = 0;
for(int i = 0; i < userCnt; i++) {
    if(userTotalScore[i] > max_num) {
        max_num = userTotalScore[i];
        max_index = i;
    }
    if(i == userCnt - 1) {
        ranking[x] = max_index;
        userTotalScore[max_index] = 0;
        max_num = 0;
        max_index = 0;
        x++;
        i = -1;
    }
    if(x == userCnt) {
        break;
    }
}
```

```
for(int i = 0; i < userCnt; i++) {
    userTotalScore[i] = Math.round(((userPoint[i] * 0.5) +
        userSpeedScore[i] + userCorrectScore[i]) * 100) / 100.0;
}
```

```
System.out.println("총점은 (점수 + 속도 + 정확도)의 환산값 입니다.");
System.out.println("이름   총점   점수   속도   정확도");
System.out.println("-----");
for(int i = 0; i < userCnt; i++) {
    System.out.println(userId[ranking[i]] + "   " + userTotalScore[ranking[i]] +
        "   " + userPoint[ranking[i]] + "   " + userSpeed[ranking[i]] +
        "   " + userCorrect[ranking[i]]);
}
```



ex) 사용자가 3명이라면

ranking = {0,0,0}

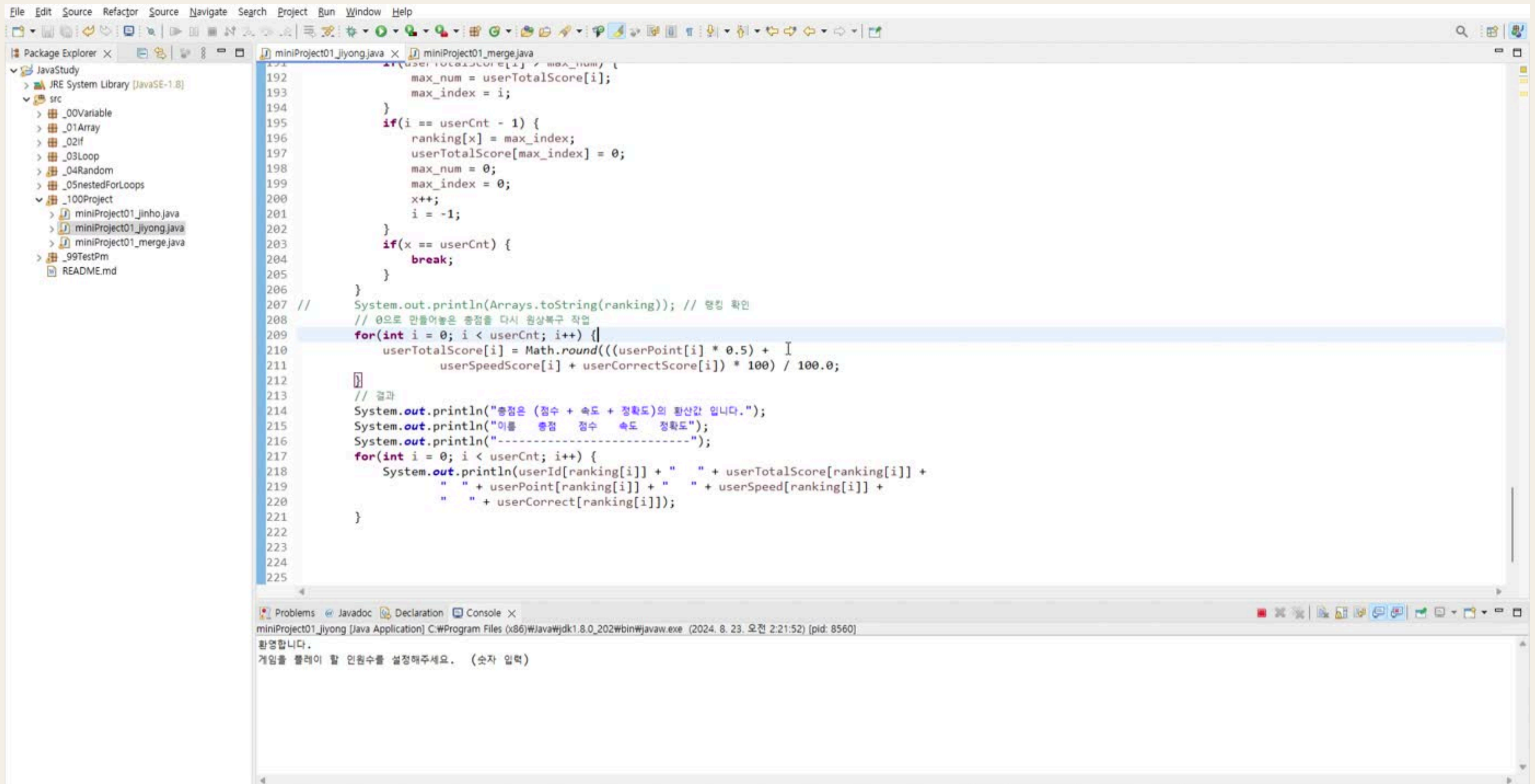
총점이 {10,30,20}으로 들어와 있으면
큰 값을 비교해 값과 인덱스를 저장하고
큰 값의 인덱스를 랭킹 배열에다가
순차적으로 저장 -> {1,2,0}

point

첫 번째 큰 값을 비교한 후에
그 다음 큰 값을 비교하기 위해서
이전 큰 값에 0을 저장해서 비교X
순위를 구하고 나면 총점을 다시 원상복구

사람 수만큼 반복문을 돌려서
총점이 높은 순서대로 통계를 출력
[ranking[0]] = > [1] => 30

시현영상



```
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JavaStudy
  JRE System Library [JavaSE-1.8]
  src
    _00Variable
    _01Array
    _02If
    _03Loop
    _04Random
    _05NestedForLoops
    _100Project
      miniProject01_jinho.java
      miniProject01_jiyong.java
      miniProject01_merge.java
    _99TestPm
    README.md

miniProject01_jiyong.java
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225

//
// 0으로 만들어놓은 총점을 다시 원상복구 작업
for(int i = 0; i < userCnt; i++) {
    userTotalScore[i] = Math.round(((userPoint[i] * 0.5) +
    userSpeedScore[i] + userCorrectScore[i]) * 100) / 100.0;
}
// 결과
System.out.println("총점은 (점수 + 속도 + 정확도)의 합산값 입니다.");
System.out.println("이름   총점   점수   속도   정확도");
System.out.println("-----");
for(int i = 0; i < userCnt; i++) {
    System.out.println(userId[ranking[i]] + "   " + userTotalScore[ranking[i]] +
    "   " + userPoint[ranking[i]] + "   " + userSpeed[ranking[i]] +
    "   " + userCorrect[ranking[i]]);
}

Problems Javadoc Declaration Console
miniProject01_jiyong [Java Application] C:\Program Files (x86)\Java\jdk1.8.0_202\bin\javaw.exe (2024. 8. 23. 오전 2:21:52) [pid: 8560]
환영합니다.
게임을 플레이 할 인원수를 설정해주세요. (숫자 입력)
```


감사합니다

2024.08.23

정지용

