



Prob48 ▾



Git:



Prob48.java ×

s Roulette {

ArrayList<String> nameLists;

String[] tmpArr;

int[] tmpIdx;

int[] success;

int nameLength;

Boolean isRedundant;

public Roulette (String[] names) {

nameLength = names.length; //스tring의 길이=숫자28

isRedundant = true;

nameLists = new ArrayList<String>(); //리스트

tmpArr = new String[nameLength]; //tmpArr=names스string

tmpIdx = new int[nameLength]; //tmpIdx=0~27까지의 숫자

success = new int[3];

int i = 0;

for (String name : names) {

tmpArr[i++] = name;

```

private Boolean checkDuplicate (int idx) { //사람이름을 무작위로 출력시, 중복없이 해주는 메소드checkDuplicate
    for (int i = 0; i < idx; i++) {
        if (tmpIdx[i] == tmpIdx[idx]) {
            return true;
        }
    }

    return false;
}
  
```

```

private Boolean checkDuplicate2 (int idx) { //당첨자3명을 무작위로 출력시, 중복없이 해주는 메소드checkDuplicate
    for (int i = 0; i < idx; i++) { //중복없이 해주는 두 메소드를 하나로 하고싶었으나, 시간이 부족해 못했
        if (success[i] == success[idx]) { //변수이름을 tmpIdx에서 success로 변경함.
            return true;
        }
    }

    return false;
}
  
```

```

public void shuffle () {
    int i = 0;

    isRedundant = true;
  
```

boulette > printSuccessArr



Prob48 ▾



Git:



Prob48.java ×

```
public void shuffle () {  
    int i = 0;  
  
    isRedundant = true;  
  
    do {  
        tmpIdx[i] = (int)(Math.random() * nameLength);  
  
        if (checkDuplicate(i)) {  
            continue;  
        }  
  
        i++;  
  
        if (i == nameLength) {  
            isRedundant = false;  
        }  
    } while (isRedundant);  
}
```

```
public void checkSuccess () {  
    int i = 0;
```

roulette > m checkSuccess



Prob48 ▾



Git:



Prob48.java ×

```
public void checkSuccess () {  
    int i = 0;  
    isRedundant = true;  
  
    do {  
        success[i] = (int)(Math.random() * nameLength);  
  
        if (checkDuplicate2(i)) {  
            continue;  
        }  
        i++;  
        if (i == 3) {  
            isRedundant = false;  
        }  
    } while (isRedundant);  
}
```

```
public void printSuccessArr () {  
  
    for (int i = 0; i < 3; i++) {  
        System.out.printf("success[%d] = %d, 당첨자{%s} 축하합니다!\n", i, success[i], tmpArr[success[i]]);  
        //출력시, tmpArr[success[i]]을 추가하여 숫자뿐만아니라 사람이름이 나오도록 했습니다.  
    }  
}
```

ette > m toString



Prob48



Git:



Prob48.java ×

@Override

```
public String toString() {  
    return "Roulette{" +  
        "tmpIdx=" + Arrays.toString(tmpIdx) +  
        "}";  
}
```

public class Prob48 {

public static void main(String[] args) {

String[] names = {

```
    "박세진", "김창욱", "김민규", "김중연", "문성호",  
    "강병화", "최승현", "유종현", "한상우", "전승리",  
    "이경환", "최준환", "김원석", "여인준", "이태양",  
    "김윤영", "정도영", "황정아", "임초롱", "김남교",  
    "이주형", "김도연", "최혜주", "김도혜", "고재권",  
    "임익환", "안보미", "이상훈"
```

};

Roulette r = new Roulette(names);

System.out.println(r);

r.shuffle();

ThirdLecture > src



Prob48



P... Test.java x Prob48.java x

ThirdLecture C:\Wj
> .idea
> out
▼ src
 ▼ Prob48.java
 Prob48
 Roulette
 Test
ThirdLecture.iml
External Libraries
Scratches and Con

```
107 "박세진", "김창욱", "김민규", "김중연", "문성호",  
108 "강병화", "최승현", "유종현", "한상우", "전승리",  
109 "이경환", "최준환", "김원석", "여인준", "이태양",  
110 "김윤영", "정도영", "황정아", "임초롱", "김남교",  
111 "이주형", "김도연", "최혜주", "김도혜", "고재권",  
112 "임익환", "안보미", "이상훈"  
113 };  
114  
115 Roulette r = new Roulette(names);  
116  
117 r.shuffle();  
118  
119 System.out.println(r);  
120  
121 r.checkSuccess();
```

Run: Prob48 x

```
↑ "C:\Program Files\ojdkbuild\java-11-openjdk-11.0.10-1\bin\java.exe" "-javaagent:C:\Users\Park sejin\Ap  
↓ Roulette{tmpIdx=[10, 16, 11, 19, 25, 1, 22, 4, 3, 9, 24, 27, 15, 21, 6, 18, 5, 17, 13, 20, 8, 2, 7, 26  
success[0] = 8, 당첨자{한상우} 축하합니다!  
success[1] = 15, 당첨자{김윤영} 축하합니다!  
success[2] = 10, 당첨자{이경환} 축하합니다!
```

Process finished with exit code 0