

```
public class CreateTeam {  
    private ArrayList<String> AteamArrayList; // AteamArrayList 변수명 ArrayList 스트링 형 배열? 생성  
    private ArrayList<String> BteamArrayList; // BteamArrayList 변수명 ArrayList 스트링 형 배열? 생성  
  
    private String[] AteamArr; // AteamArrayList -> nameA{...} 이름이 9개 들어간 AteamArr 배열 생성  
    private String[] BteamArr; // BteamArrayList -> nameB{...} 이름이 8개 들어간 BteamArr 배열 생성  
  
    private int AnumOfPerson; // AnumOfPerson -> 9  
    private int BnumOfPerson; // BnumOfPerson -> 8  
  
    private int numOfTeam; // TEAMNUMBER , 2  
  
    public CreateTeam(String[] arrA, String[] arrB, final int TEAMNUMBER) {  
        // 생성자 CreateTeam 생성, 매개변수 arrA배열(nameA), arrB배열(nameB), TEAMNUMBER(2)  
        AteamArr = arrA; // AteamArr 에 arrA 대입  
        BteamArr = arrB; // BteamArr 에 arrB 대입  
  
        AnumOfPerson = arrA.length; // AnumOfPerson 변수에 arrA.length 길이 만큼 대입 9  
        BnumOfPerson = arrB.length; // BnumOfPerson 변수에 arrB.length 길이 만큼 대입 8  
  
        numOfTeam = TEAMNUMBER; // A조 B조 2개  
  
        AteamArrayList = new ArrayList<String>(); // 현재 매서드 AteamArrayList 는 아무 값도 할당되어 있지 않음  
        BteamArrayList = new ArrayList<String>();  
    }  
}
```

```
public void allocRandomTeam() { // 랜덤으로 팀을 할당하는 메서드
```

```
// 1) 배열을 백업 받았으므로 랜덤값을 뽑아서
```

```
// 해당 값을 인덱스로 특정 인물을 추출하여 ArrayList에 설정한다.
```

```
/*
```

```
for(int i = 0; i < AnumOfPerson; i++) {
```

```
int randNum = (int)(Math.random() * AnumOfPerson);
```

```
AteamArrayList.add(AteamArr[randNum]);
```

```
}
```

```
for(int i = 0; i < BnumOfPerson; i++) {
```

```
int randNum = (int)(Math.random() * BnumOfPerson) + AnumOfPerson;
```

```
BteamArrayList.add(BteamArr[randNum]);
```

```
}
```

```
*/
```

```
allocArrayList(AteamArrayList, AteamArr, AnumOfPerson); // 현재 allocArrayList(빈 값, A팀 배열 값, A팀 length값 9)
```

```
allocArrayList(BteamArrayList, BteamArr, BnumOfPerson);
```

```
}
```

```

public void allocArrayList(
    ArrayList<String> al, // AteamArrayList al
    String[] arr, // AteamArr[] arr
    int loopNum) { // 9
    // loopNum 은 AnumOfPerson 9 혹은 BnumOfPerson 8 배열의 길이 값이 맞나요?
    // name A { ... } 배열의 수만큼 인가요??

    boolean isDup = false; // isDup 을 false 로 두는 이유 : isDup = true; 가 될 경우 continue;를 하기 위해??
    // 아니면 처음은 아무것도 할당된 것이 없으니 무조건 한번은 실행 하기 위해서??

    for (int i = 0; i < loopNum; i++) { // i 가 loopNum 을 도는 동안 (AnumOfPerson, A조의 경우 9)

        do {
            // do-while 문: 무조건 한번은 실행
            int randNum = (int) (Math.random() * loopNum);
            // randNum 에 Math.random 값 * 9 = 0~8 (0.9999 * 9 = 8.1 -> 0 ~ 8)

            if (al.contains(arr[randNum])) { // 할당된 인덱스에 문자열의 중복여부 체크
                isDup = true; // isDup 이 true; 중복이라면 for 문 부터 루프를 다시 돌
                continue;
            } else {
                isDup = false; // 중복이 아니라면 false 로 al.add(arr[randNum]) 에 이름을 추가해줌
            }

            al.add(arr[randNum]); // al.add 는 al에 이름을 추가해주는건 알겠는데
                                // (arr[randNum]) 이걸 무엇을 뜻하나요? 랜덤으로 할당된 인덱스 번호의 배열인가요?

        } while (isDup); // isDup 이 false 일 동안 = 0~8 인덱스에 전부 할당이 될 동안인가요?
    }
}

```

```
while (e.hasNext()) {  
    // 존재하는 값을 가져와서 Integer 형식으로 저장합니다.  
    name = (String) e.next();  
    System.out.printf("%s ", name);  
    // hasNext ...란 무엇일까...?  
    // boolean 자료형 타입으로 반환됨 True or False  
    // 현재 케이스에 어떻게 쓰이는지 모르겠다...  
  
    // 현재 케이스에서는 무조건 앞에 5명이 나온다.  
    // 그러므로 이것도 랜덤하게 4, 5 혹은 5, 4가 나오게 해줘야 한다.  
    if (needException) {  
        randValue = (int) (Math.random() * 2); //  
        needException = false;  
    }  
    if ((cnt % (quot + randValue)) == 0) {  
        System.out.println("");  
  
        if (cnt == 4) {  
            randValue = 1;  
        } else {  
            randValue = 0;  
        }  
        cnt = 0;  
    }  
    cnt++;  
}
```