```java
package Fourteenth;

import java.util.ArrayList;
import java.util.Iterator;

public class CreateTeam {
    private ArrayList<String> AteamArrayList;
    private ArrayList<String> BteamArrayList;

    private String[] AteamArr;
    private String[] BteamArr;

    private int AnumOfPerson;
    private int numOfTeam;

    private int BnumOfPerson;

    public CreateTeam(String[] arrA, String[] arrB, final int TEAMNUMBER) {
        AteamArr = arrA; (9명)
        BteamArr = arrB; (8명)

        AnumOfPerson = arrA.length; 9
        BnumOfPerson = arrB.length; 8

        numOfTeam = TEAMNUMBER; 2

        AteamArrayList = new ArrayList<String>();
        BteamArrayList = new ArrayList<String>();
    }

    public void allocRandomTeam() {
        // 문제 풀 때 어떤식으로 접근을 해야할까요 ?
        // 1. 우선 이 문제를 풀기 위한 전략을 세워야 한다.
        // 2. 문제가 요구 사항을 파악해야한다.
        // 3. 요구 사항을 하나 하나 각개 격파한다.

        // 위의 방식을 따라서 현재 문제에 대한 요구 사항을 파악해보자!
        // 문제: 우리반 사람들을 4개의 팀으로 랜덤하게 할당한다.
        // 1) 랜덤을 사용해야 한다(Math.random() 혹은 Random 클래스)
```

10명을 아, 11명을 n개의 팀으로 나누기.

현재 팀 2개의 배열을 1개의 배열로 만들기 — 랜덤값이.. OK.

팀원 수 % n개의팀 == 0 → n개의팀으로 나누기. — 나눈값이상 같은명 만들어지기. n개로 나누어 떨어지면 \n 하기. OK.

팀원 수 % n개의팀 > 0 → 랜덤으로 — 한개의팀 혹은 주기.?

? ⇒ +1 해준다?

A조 배열 B조 배열

생성자 통해 값받기.

현재는 비어있음.

1이두개를 포함하는 method 만들기 CNT 써서 그대로 하기.

```
        allocArrayList(
                AteamArrayList, AteamArr, AnumOfPerson
        );
        allocArrayList(
                BteamArrayList, BteamArr, BnumOfPerson
        );
    }

    public void allocArrayList(
            ArrayList<String> al,
            String[] arr,
            int loopNum) {

        boolean isDup = false;

        for(int i = 0; i < loopNum; i++) {
            // 우리는 A 배열, B 배열이 나눠진 케이스를 받으므로
            // 하나의 배열에서 모든 값을 처리하지 않는다.
            // 그러므로 start를 별도로 만들 필요가 없었다.
            // int randNum = (int)(Math.random() * loopNum) + start;
            do {
                int randNum = (int)(Math.random() * loopNum);

                if(al.contains(arr[randNum])) {
                    isDup = true;
                    continue;
                } else {
                    isDup = false;
                }

                al.add(arr[randNum]);
            } while(isDup);
        }
    }
```

이 method에서 만든 값들이 위로 올라감.

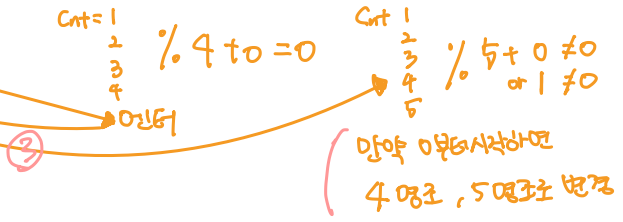
중복없이 랜덤값 위한 메서드.

중복값 없을때까지 무한 반복순환해?

리턴값

```java
public void printArrayList(ArrayList<String> al) {
    String name;
    // ticketArrayList를 순회할 수 있는 정보를 얻음
    Iterator e = al.iterator();

    int cnt = 1;
    int divNum;
    int quot = al.size() / numOfTeam;
    int remain = al.size() % numOfTeam;
    boolean needException =
            (remain > 0) ? true : false;

    int randValue = 0;
    // 현재 발생하는 문제
    // 1) 매 반복마다 randValue가 0, 1, 0, 1 스위칭되고 있음

    // 순회할 수 있는가 ?
    // 데이터가 없으면 루프 진행 x
    // 데이터가 하나라도 있으면 루프 진행 o
    while(e.hasNext()) {
        // 존재하는 값을 가져와서 Integer 형식으로 저장합니다.
        name = (String) e.next();
        System.out.printf("%s ", name);

        // 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;
        }

        /*
        System.out.printf("cnt = %d, quot + randValue = %d\n",
                cnt, quot + randValue);
         */
```

```
148             if((cnt % (quot + randValue)) == 0) {
149                 System.out.println("");
150 
151                 if(cnt == 4) {
152                     randValue = 1;
153                 } else {
154                     randValue = 0;
155                 }
156 
157                 cnt = 0;
158             }
159 
160             cnt++;
161         }
162 
163         System.out.println("");
164     }
165 
166     public ArrayList<String> getAteamArrayList() {
167         return AteamArrayList;
168     }
169 
170     public ArrayList<String> getBteamArrayList() {
171         return BteamArrayList;
172     }
173 }
```



```java
package Fourteenth;

public class AllocTeamHomework {
    public static void main(String[] args) {
        final int TEAMNUMBER = 2;
        System.out.println("지금부터 랜덤 팀 구성을 시작합니다.");

        String[] nameA = {
                "고동영", "장해솔", "류슬기", "박재민", "한다은",
                "최현정", "오진욱", "조진형", "이정현"
        };
        String[] nameB = {
                "이범진", "박소현", "탁성진", "노찬욱",
                "박기범", "하진주", "이승윤", "최임식"
        };

        // 단순히 고정된 사람들을 받는 작업
        // CreateTeam ct = new CreateTeam(name);
        CreateTeam ct = new CreateTeam(nameA, nameB, TEAMNUMBER);

        ct.allocRandomTeam();
        ct.printArrayList(ct.getAteamArrayList());
        ct.printArrayList(ct.getBteamArrayList());
    }
}
```