

Report_Chap1 1

Date: 2022. 06. 07

Name: 송예지

Student ID: 22100396

Given the definition of a 2D array such as the following:

```
String[][] data = {{("A","B"),  
                    {"1","2"},  
                    {"XX","YY","ZZ"}}};
```

Write a recursive Java program that outputs all combinations of each subarray in order. In the above example the desired output (although it doesn't have to be in this order) is:

```
A 1 XX  
A 1 YY  
A 1 ZZ  
A 2 ZZ  
A 2 YY  
A 2 ZZ  
B 1 XX  
B 1 YY  
B 1 ZZ  
B 2 ZZ  
B 2 YY  
B 2 ZZ
```

Your program should work with arbitrarily sized arrays in either dimension.

For example, the following data:

```
String[][] data = {{("A"),  
                    {"1"},  
                    {"2"},  
                    {"XX","YY"}}};
```

should output:

```
A 1 2 XX  
A 1 2 YY
```

```
String[][] data = {{("A","B"),  
                    {"1","3"},  
                    {"2"},  
                    {"XX","YY"}}};
```

다음과 같은 2D 어레이의 정의를 제공합니다.

```
String[][] data = {{("A","B"),  
                    {"1","2"},  
                    {"XX","YY","ZZ"}}};
```

각 하위 배열의 모든 조합을 순서대로 출력하는 재귀 자바 프로그램을 작성한다. 위의 예에서 원하는 출력(이 순서일 필요는 없지만)은 다음과 같습니다.

```
A 1 XX  
A 1 YY  
A 1 ZZ  
A 2 ZZ  
A 2 YY  
A 2 ZZ  
B 1 XX  
B 1 YY
```

B 1 ZZ
B 2 ZZ
B 2 YY
B 2 ZZ

프로그램은 임의의 크기의 배열과 함께 작동해야 합니다.

예를 들어 다음과 같은 데이터가 있습니다.

```
String[][] data = {"A",  
                   {"1",  
                    {"2",  
                     {"XX", "YY"}}};
```

결과는:

A 1 2 XX
A 1 2 YY

```
String[][] data = {"A", "B",  
                   {"1", "3",  
                    {"2",  
                     {"XX", "YY"}}};
```

기본 문제 해결

- ⇒ Recursion 함수의 parameter 는 3 개
 - String[][] array
 - 시작 array row 의 index
 - 문자열을 저장할 수 있는 변수
- ⇒ 함수를 끝낼 수 있는 조건: 자신을 부르는 횟수가 array row 의 length 와 같아지면 종료
- ⇒ For 문을 array n 번째의 column 길이만큼 반복
 - 문자열을 저장할 수 있는 local 변수 생성
 - Parameter 로 들어온 문자열에 아무것도 존재하지 않으면 그대로 array[시작 Index][for 문의 i]을 새로 생성한 문자열 변수에 담기
 - Parameter 로 들어온 문자열에 문자가 존재한다면 들어온 문자와 array[시작 index][for 문의 i]을 새로 생성한 문자열 변수에 담기
 - 자기 함수 호출
 - > String[][] array
 - > array row 의 index 에서 다음 row 의 index 값
 - > for 문에서 생성한 문자열 parameter 로 넘기기

실습 코드

```
public class StringArrayPrint {  
    public static void main(String args[]){  
        String s="";  
        String[][] data1 = {{ "A", "B",  
                                {"1", "2"},  
                                {"XX", "YY", "ZZ"}}};  
  
        String[][] data2 = {{ "A",  
                                {"1"},  
                                {"2"},  
                                {"XX", "YY"}}};  
  
        String[][] data3 = {{ "A", "B",  
                                {"1", "3"},  
                                {"2"},  
                                {"XX", "YY"}}};  
  
        print_data(data1, 0, s);  
        System.out.println("");  
        print_data(data2, 0, s);  
    }  
}
```

```
public class StringArrayPrint {  
    public static void main(String agrs[]){
```

```
String s="";
String[][] data1 = {{ "A", "B"},
                    {"1", "2"},
                    {"XX", "YY", "ZZ"}};
```

→ Data 1

```
String[][] data2 = {{ "A"},
                    {"1"},
                    {"2"},
                    {"XX", "YY"}};
```

→ Data 2

```
String[][] data3 = {{ "A", "B"},
                    {"1", "3"},
                    {"2"},
                    {"XX", "YY"}};
```

→ Data 3

```
print_data(data1, 0, s);
```

→ data 1 출력

```
System.out.println("");
print_data(data2, 0, s);
```

→ data 2 출력

```
System.out.println("");
print_data(data3, 0, s);
```

→ data 3 출력

```
System.out.println("");
```

```
}
```

```
public static void print_data(String[][] list, int n, String s) {
```

```
    if(n == list.length) {
```

→ parameter로 들어온 N의 값이 list row의 길이와 같아면

```
        System.out.println(s);
```

→ Parameter로 들어온 s를 출력

```
        return;
```

→ 함수 종료

```
    }
```

```
    for(int i = 0; i<list[n].length; i++) {
```

→ List n번째 column의 값만큼 반복문 실행

```
        String result;
```

→ 빈 string variable 생성

```
        if(s == "") result = list[n][i];
```

→ S에 값이 없으면 list[n][i] 를 result에 담기

```
        else result = s + " " + list[n][i];
```

→ S에 값이 존재하면 s 뒤에 list[n][i]번째 값 붙이기

```
        print_data(list, n+1, result);
```

➔ 자기 자신 함수 호출 (recursion)

}

}