

**CSE2016**

**프로그램설계방법론**

# **자료 구조. 배열**

**Data Structure : Array**

**도경구**

한양대학교 ERICA 소프트웨어학부



# Array

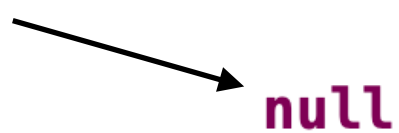
동일한 타입의 값을 정해진 개수만큼 나란히 모아놓은 구조

```
int[] r;
```

```
int[] r → null
```

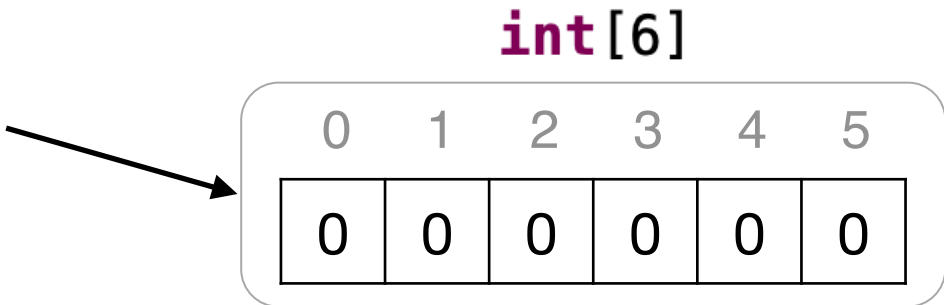
```
int[] r;
```

```
int[] r
```



```
int[] r = new int[6];
```

```
int[] r
```



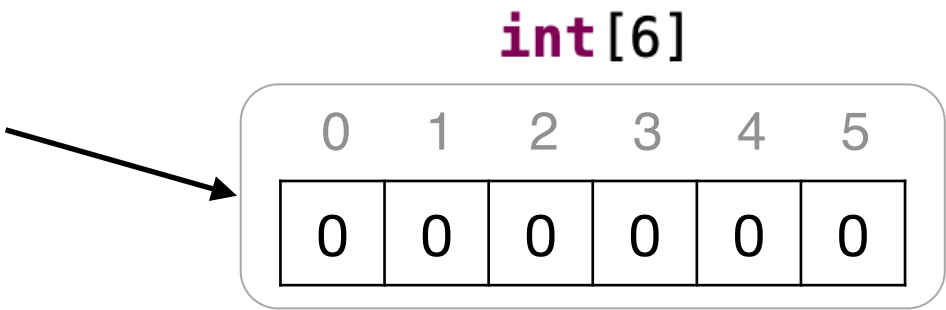
- 배열array은 객체object
- 배열 생성시 각 원소는 기본값으로 초기화

타입	초기값
int	0
double	0.0
boolean	false
Object	null

- 인덱스index는 정수로 0부터 시작
- 배열의 크기가 6이면, 인덱스의 범위는 0~5

```
int[] r = new int[6];
```

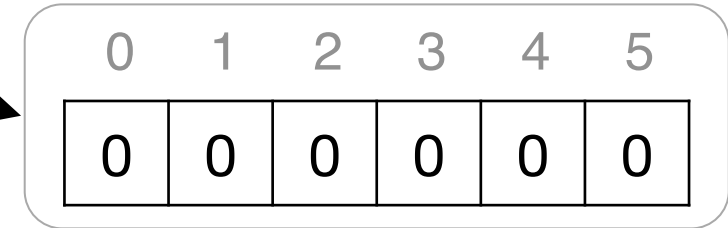
```
int[] r
```



```
int[] r = new int[6];
```

```
int[] r
```

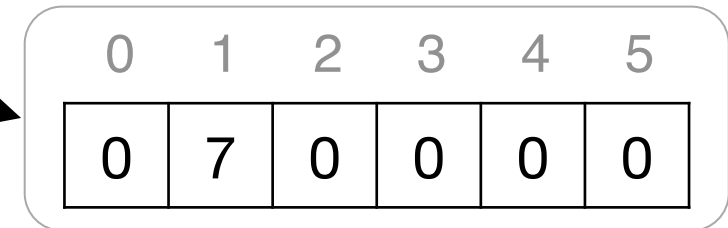
```
int[6]
```



```
r[1] = 7;
```

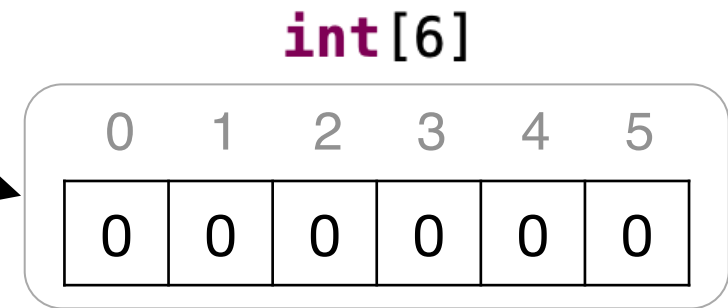
```
int[] r
```

```
int[6]
```



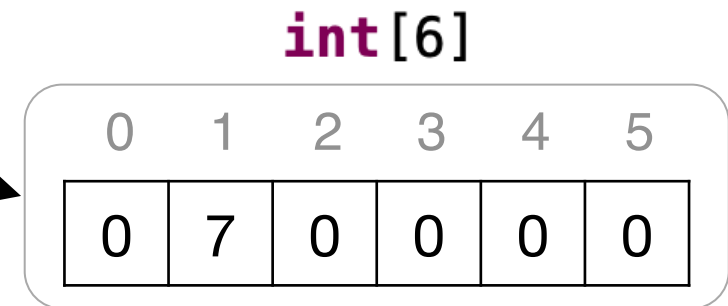
```
int[] r = new int[6];
```

```
int[] r
```



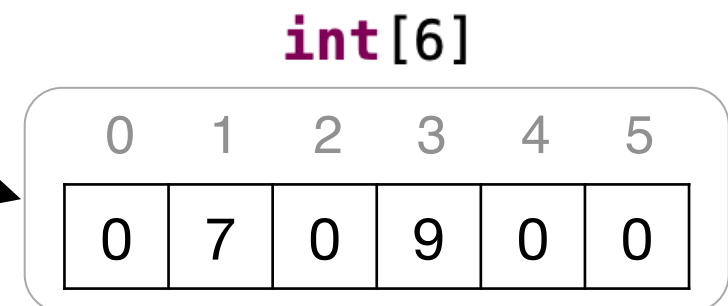
```
r[1] = 7;
```

```
int[] r
```



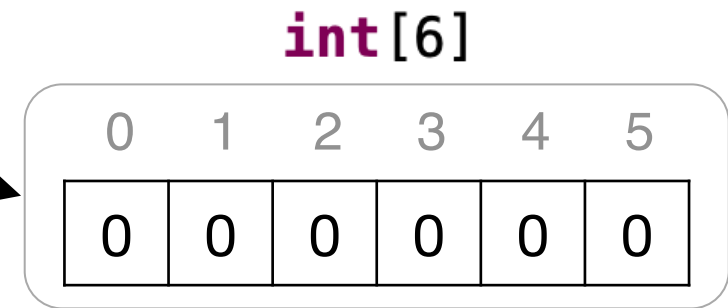
```
r[3] = r[1] + 2;
```

```
int[] r
```



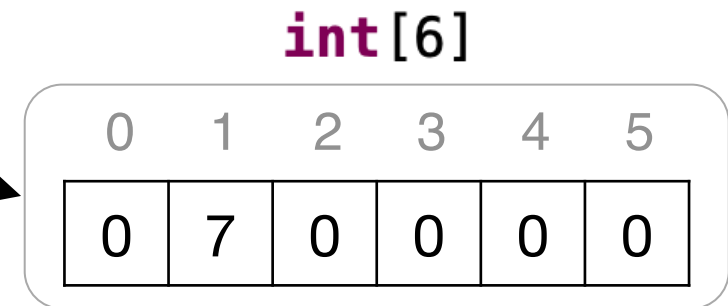
```
int[] r = new int[6];
```

```
int[] r
```



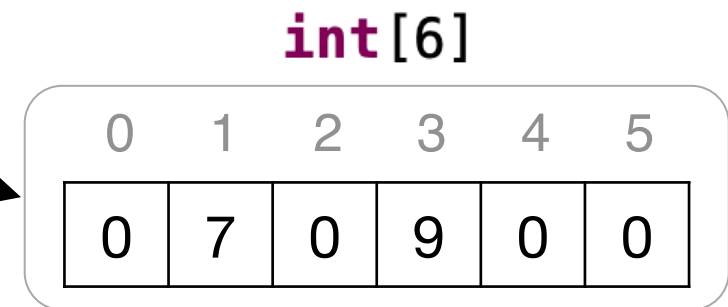
```
r[1] = 7;
```

```
int[] r
```



```
r[3] = r[1] + 2;
```

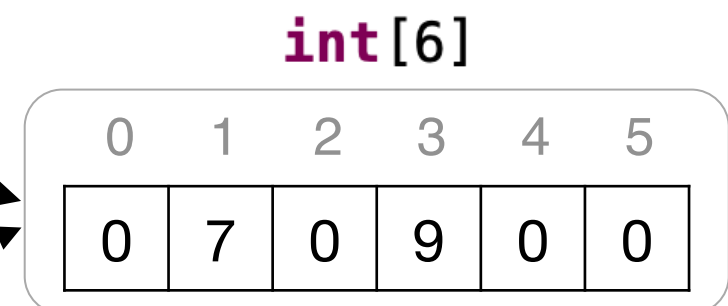
```
int[] r
```



```
int[] s = r;
```

```
int[] r
```

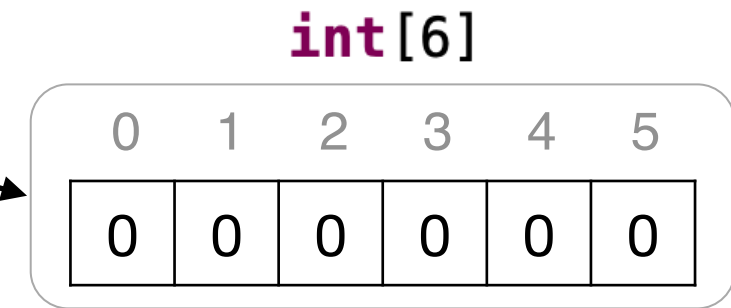
```
int[] s
```





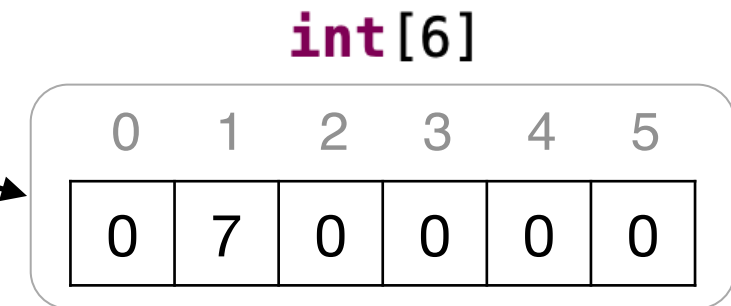
```
int[] r = new int[6];
```

```
int[] r
```



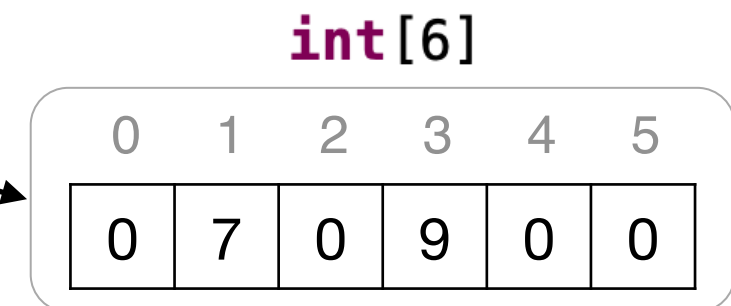
```
r[1] = 7;
```

```
int[] r
```



```
r[3] = r[1] + 2;
```

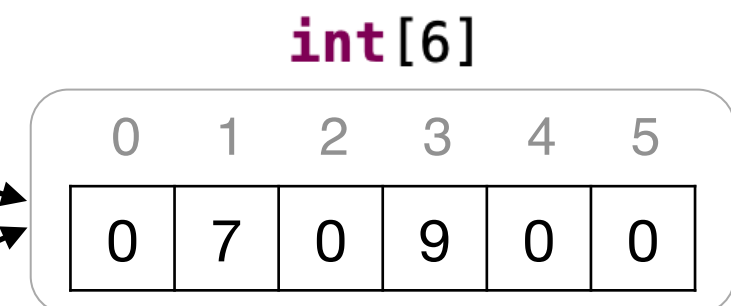
```
int[] r
```



```
int[] s = r;
```

```
int[] r
```

```
int[] s
```



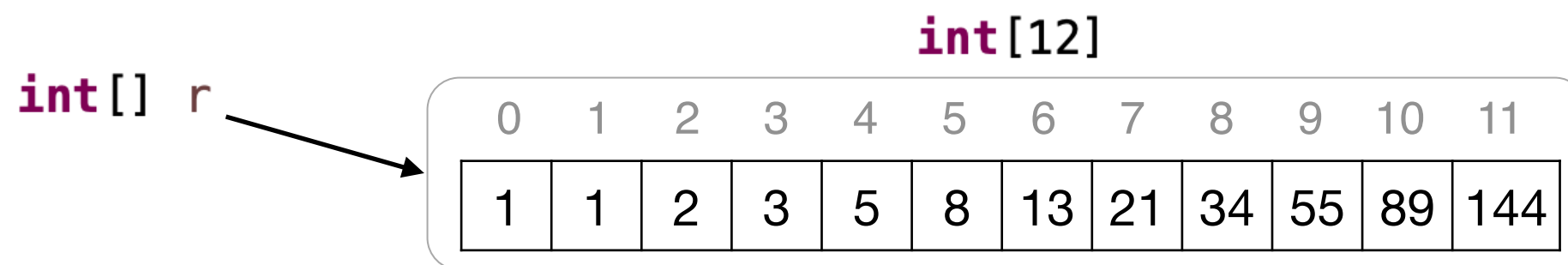
- 주의** : 실행 중 범위를 벗어난 인덱스로 배열을 참조하면 `ArrayIndexOutOfBoundsException` 오류 발생

## 배열 생성 + 값 채우기

```
int[] r = new int[12];  
r[0] = 1;  
r[1] = 1;  
for (int i = 2; i < r.length; i = i + 1) {  
    r[i] = r[i-1] + r[i-2];  
}
```

## 배열 생성 + 값 채우기

```
int[] r = new int[12];  
r[0] = 1;  
r[1] = 1;  
for (int i = 2; i < r.length; i = i + 1) {  
    r[i] = r[i-1] + r[i-2];  
}
```

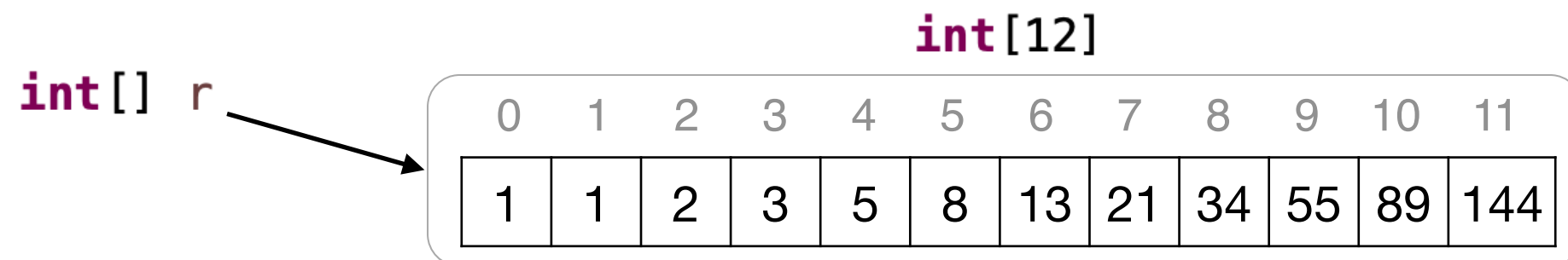


## 배열 생성 지름길

```
int[] r = {1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144};
```

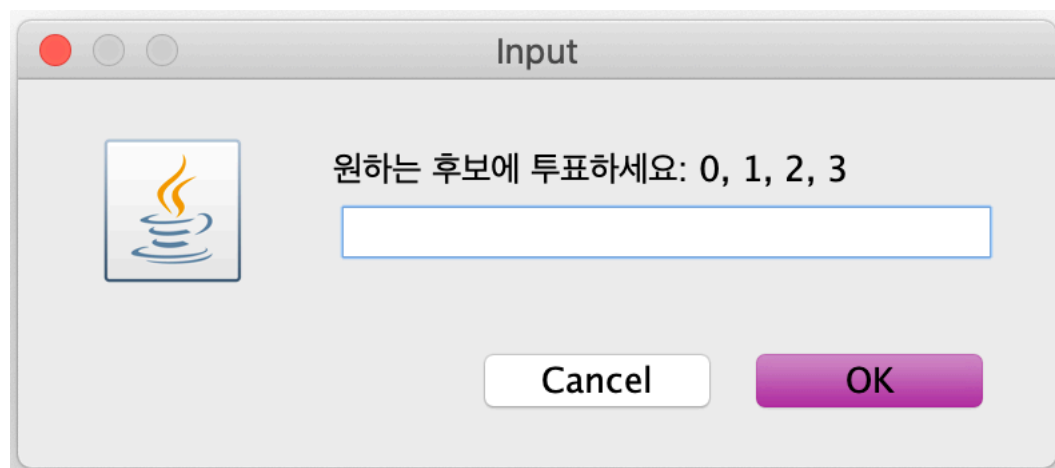
## 배열 생성 지름길

```
int[] r = {1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144};
```



# Collecting Input Data in Arrays

## 사례 - 투표 애플리케이션



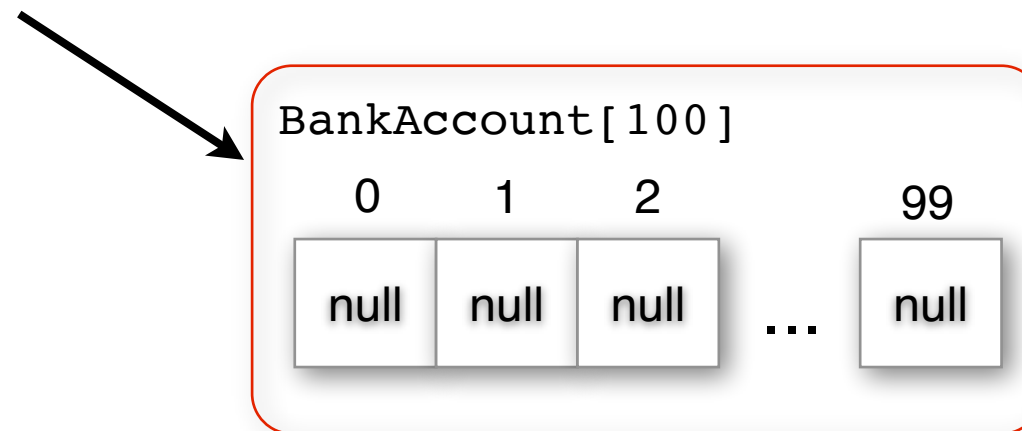
후보	0번이	4표를	득표하였습니다.
후보	1번이	8표를	득표하였습니다.
후보	2번이	2표를	득표하였습니다.
후보	3번이	3표를	득표하였습니다.

```
1 import javax.swing.*;
2
3 /** VoteCount 투표 애플리케이션 */
4 public class VoteCount {
5
6     public static void main(String[] args) {
7         int num_candidates = 4;
8         int[] votes = new int[num_candidates]; // 0으로 자동 초기화
9
10        boolean processing = true;
11        while (processing) {
12            /* loop invariant: 투표는 해당 배열에 누적되었음 */
13            String input = JOptionPane.showInputDialog("원하는 후보에 투표하세요: 0, 1, 2, 3");
14            int v = Integer.parseInt(input);
15            if (v >= 0 && v < votes.length)
16                votes[v] = votes[v] + 1;
17            else
18                processing = false;
19        }
20        for (int i = 0; i != votes.length; i = i + 1)
21            System.out.println("후보 " + i + "번이 " + votes[i] + "표를 득표하였습니다.");
22    }
23 }
```

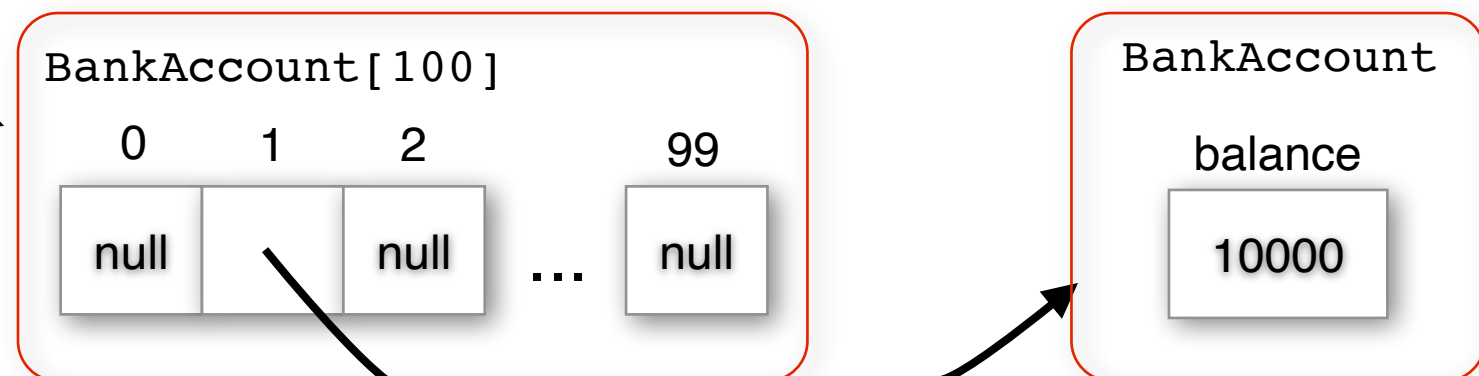
# Array of Objects

## 사례 - 은행계좌 관리 애플리케이션

```
BankAccount[] bank = new BankAccount[100];
```



```
bank[1] = new BankAccount(10000);
```

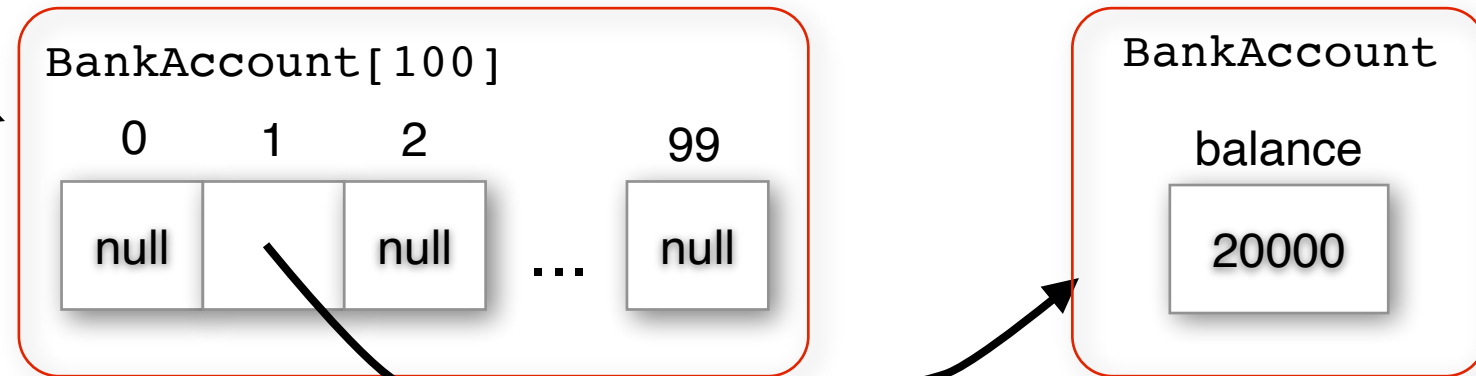




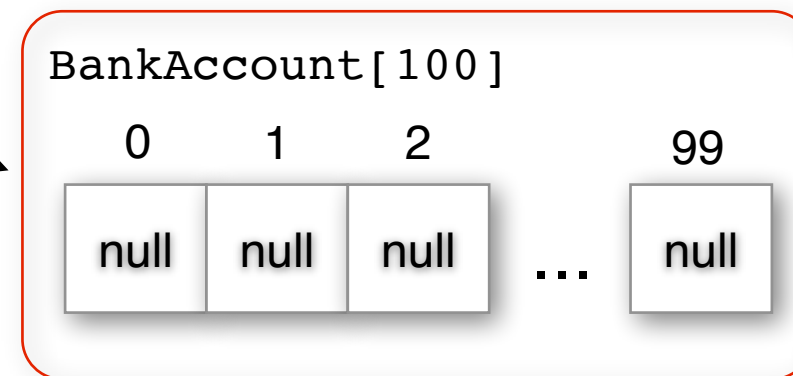
# Array of Objects

## 사례 - 은행계좌 관리 애플리케이션

```
bank[1].deposit(10000);
```



```
bank[1] = null
```



## 사례 학습 - 카드 게임

- 카드
  - 무늬 : Diamonds, Hearts, Clubs, Spades
  - 등급 : A, 2~10, Jack, Queen, King
- 카드 덱
  - 카드 배열



# Specification

class	CardDeck	카드 1벌
field	Card[] deck	남은 카드
method	Card newCard()	카드 덱에서 임의의 새 카드 한 장을 뽑아 리턴, 없으면 카드 1벌을 새로 만들고 한 장을 뽑아 리턴
	boolean moreCard()	카드 덱에 카드가 남아있는지 알려줌 있으면 true를 리턴, 없으면 false를 리턴

class	Card	카드 1장
field	String suit	카드의 무늬
	int rank	카드의 등급
method	String getSuit()	카드의 무늬 리턴
	int getRank()	카드의 등급 리턴

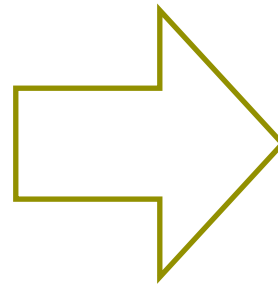
CardDeck, Card 클래스를 활용하여 다음 명세를 만족하는 Hand 클래스를 만들자.

class	Card	카드 놀이 하는 참여자의 손에 들고 있는 카드의 모델
field	Card[] <b>hand</b>	손에 들고 있는 카드
	int <b>number_of_cards</b>	손에 들고 있는 카드의 장수
method	void <b>showHand()</b>	들고 있는 카드를 모두 실행창에 프린트한다. 카드가 없으면 카드가 없다고 프린트한다.
	boolean <b>receiveCard</b> (Card c)	카드를 한 장 받는다. 한도를 초과하면 받을 수 없다
	Card <b>removeCard()</b>	들고 있는 카드 중에서 하나를 버린다. 게임에 따라 버릴 카드를 선택하는 기준이 다를 수 있음. 여기서는 무작위로 선택 손에 들고 있는 카드가 없으면 null을 리턴

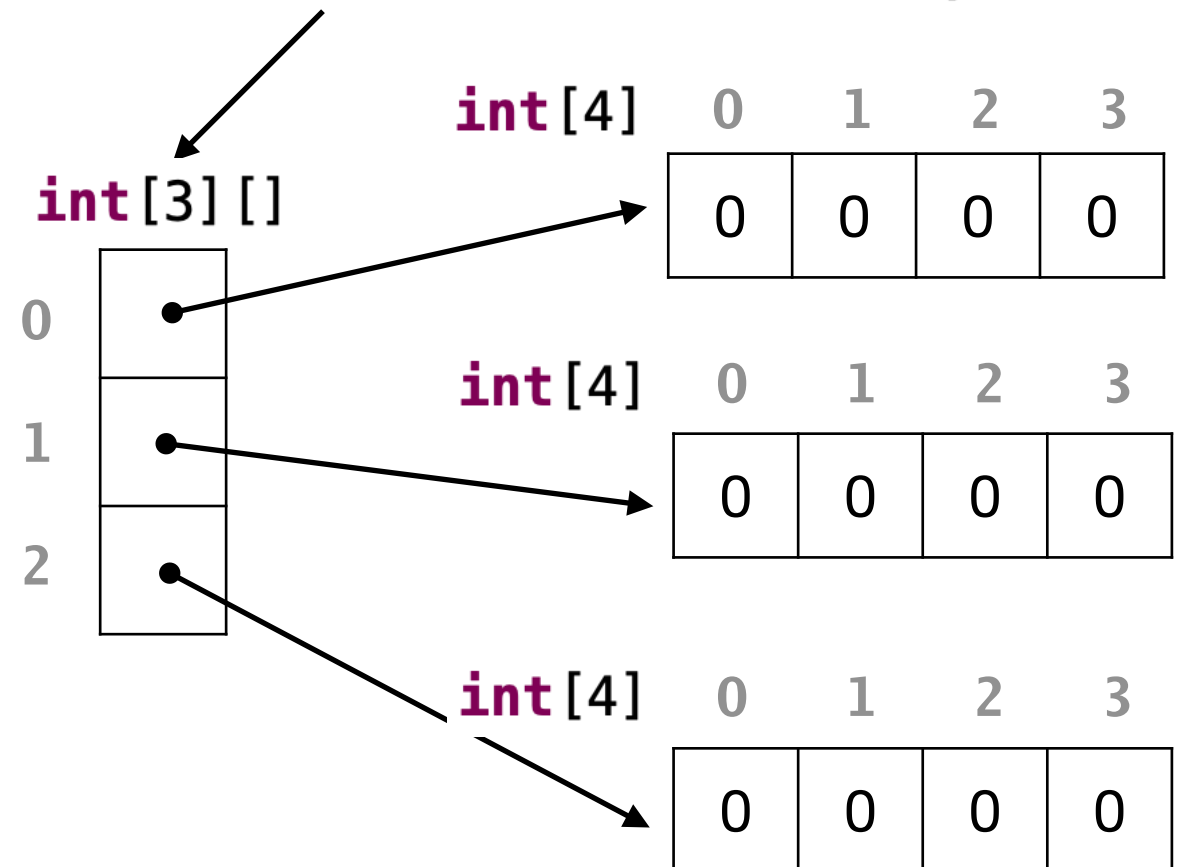
# Two-Dimensional Arrays 2차원 배열

## 사례 : 학생회장 선거 득표 현황

	기 호 1 번	기 호 2 번	기 호 3 번	기 호 4 번
	0	1	2	3
1학년	0			
2학년	1			
3학년	2			



```
int[][] a = new int[3][4];
```



```
a[1][2] = a[1][2] + 1
```

- 가로줄(행row)의 수 : `a.length`
- 세로줄(열column)의 수 : `a[0].length`

## Two-Dimensional Arrays 2차원 배열

사례 : 학생회장 선거 득표 현황

기 호 1 번	기 호 2 번	기 호 3 번	기 호 4 번
0	1	2	3

1학년 0

2학년 1

3학년 2

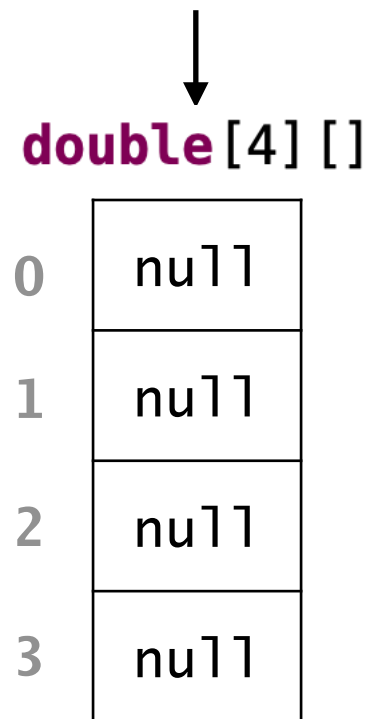

```
int[][] election = new int[3][4];
```

```
for (int j = 0; j < 4; j++) {
    int votes = 0;
    for (int i = 0; i < 3; i++)
        votes = votes + election[i][j];
    System.out.println("기호" + (j+1) + "번은 " + votes + "표 받았습니다.");
}
```

```
for (int i = 0; i < 3; i++) {
    int votes = 0;
    for (int j = 0; j < 4; j++)
        votes = votes + election[i][j];
    System.out.println((i+1) + "학년은 " + votes + "표 행사했습니다.");
}
```

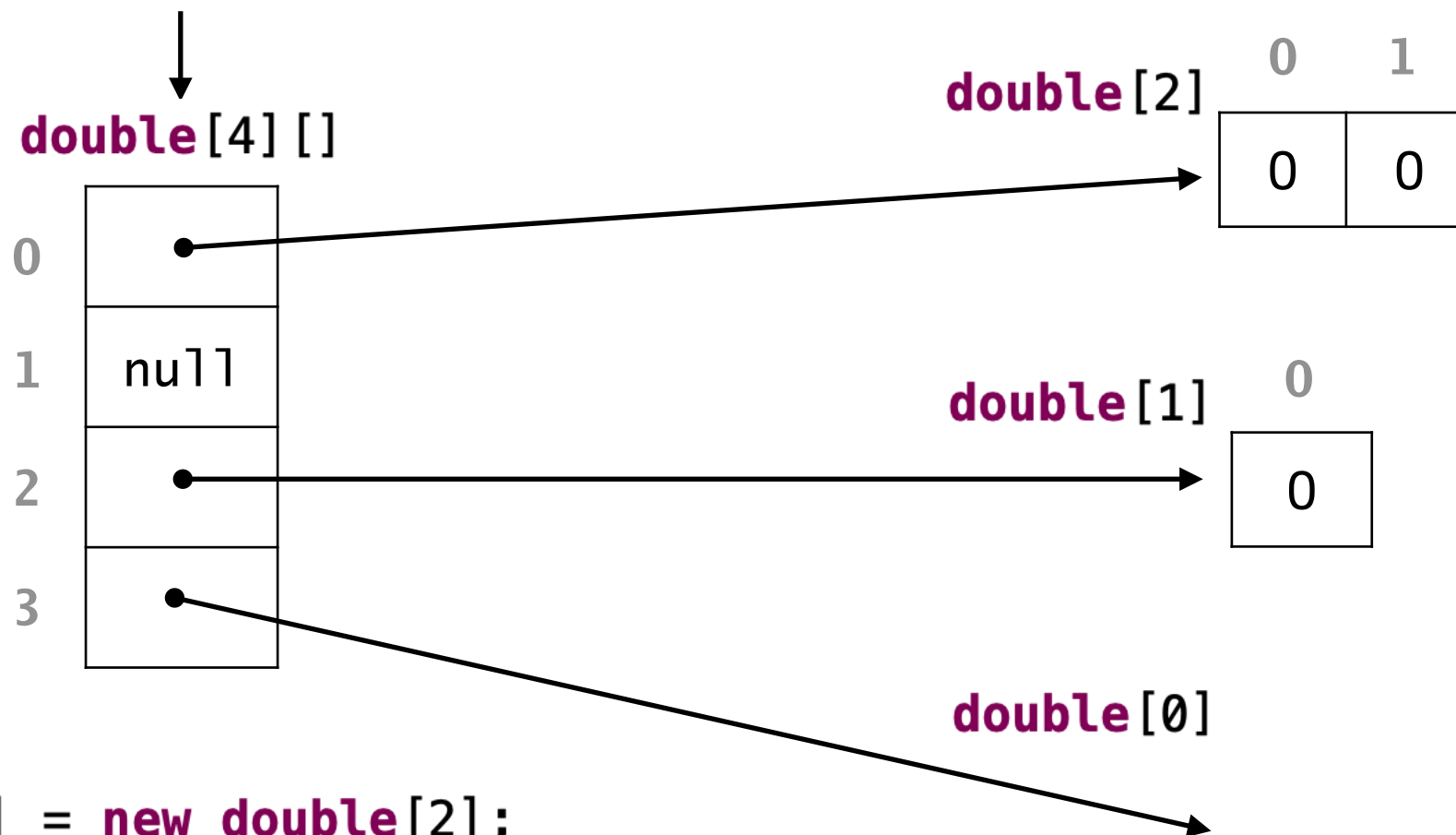
# Ragged Arrays    들쭉날쭉 배열

```
double[][] ragged = new double[4][];
```



# Ragged Arrays 들쭉날쭉 배열

```
double[][] ragged = new double[4][];
```



```
double[0] = new double[2];
double[2] = new double[1];
double[3] = new double[0];
```



# Ragged Arrays 들쭉날쭉 배열

```
public static void main(String[] args) {

    int max words = 20;
    char[][] word = new char[max_words][];
    int count = 0;
    boolean processing = true;
    while (processing && count < max_words) {
        String s = JOptionPane.showInputDialog("단어를 입력하세요.");
        if (s.equals(""))
            processing = false;
        else {
            word[count] = new char[s.length()];
            for (int i = 0; i < s.length(); i++)
                word[count][i] = s.charAt(i);
            count++;
        }
    }
}
```

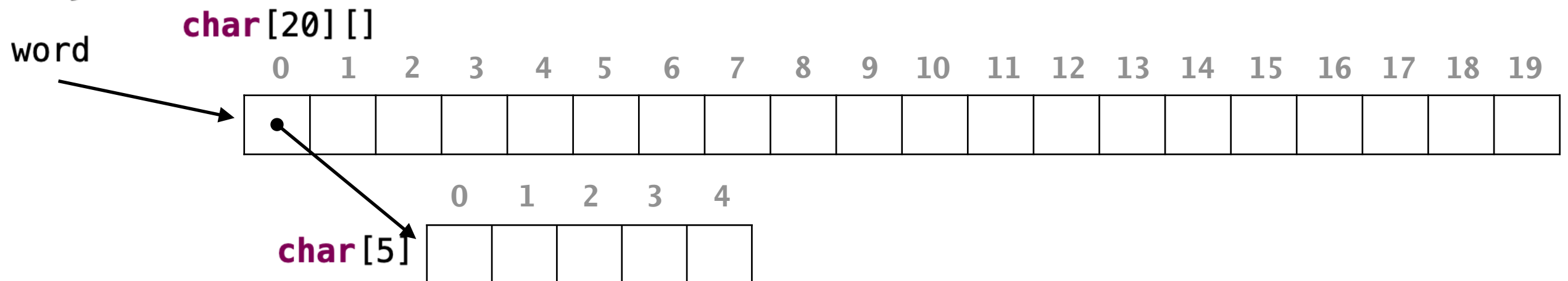


# Ragged Arrays 들쭉날쭉 배열

```
public static void main(String[] args) {

    int max_words = 20;
    char[][] word = new char[max_words][];
    int count = 0;
    boolean processing = true;
    while (processing && count < max_words) {
        String s = JOptionPane.showInputDialog("단어를 입력하세요.");
        if (s.equals(""))
            processing = false;
        else {
            word[count] = new char[s.length()];
            for (int i = 0; i < s.length(); i++)
                word[count][i] = s.charAt(i);
            count++;
        }
    }
}
```

`s = "sound";`



# Ragged Arrays 들쭉날쭉 배열

```
public static void main(String[] args) {
```

```
    int max_words = 20;
```

```
    char[][] word = new char[max_words][];
```

```
    int count = 0;
```

```
    boolean processing = true;
```

```
    while (processing && count < max_words) {
```

```
        String s = JOptionPane.showInputDialog("단어를 입력하세요.");
```

```
        if (s.equals(""))
```

```
            processing = false;
```

```
        else {
```

```
            word[count] = new char[s.length()];
```

```
            for (int i = 0; i < s.length(); i++)
```

```
                word[count][i] = s.charAt(i);
```

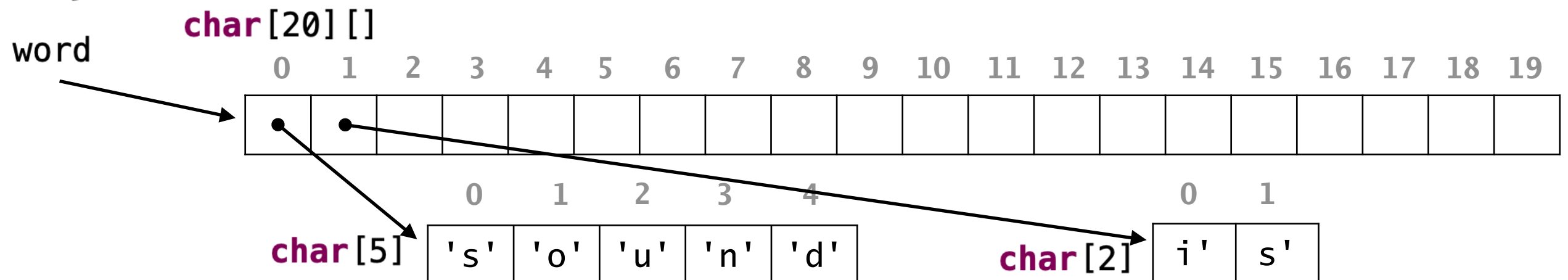
```
            count++;
```

```
        }
```

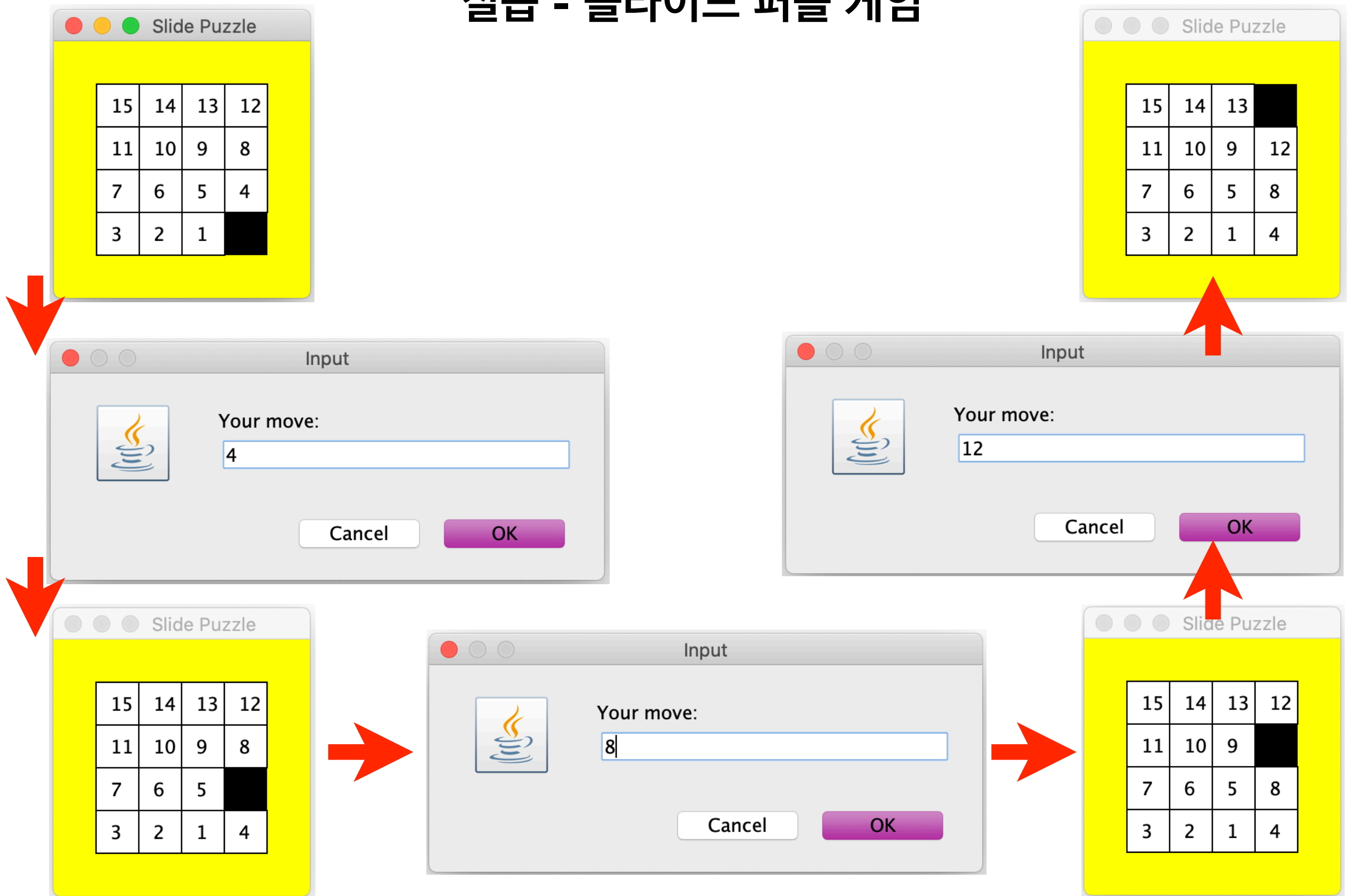
```
    }
```

```
s = "sound";
```

```
s = "is";
```



## 실습 - 슬라이드 퍼즐 게임



## 실습 - 슬라이드 퍼즐 게임

