

자료 구조. 배열

Data Structure : Array



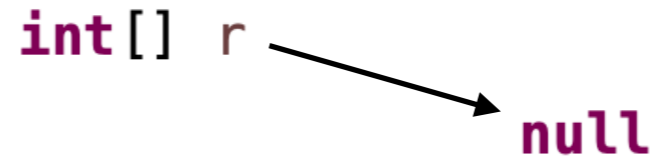
배열 Array

동일한 타입의 값을 정해진 개수만큼 나란히 이어 붙여 놓은 구조

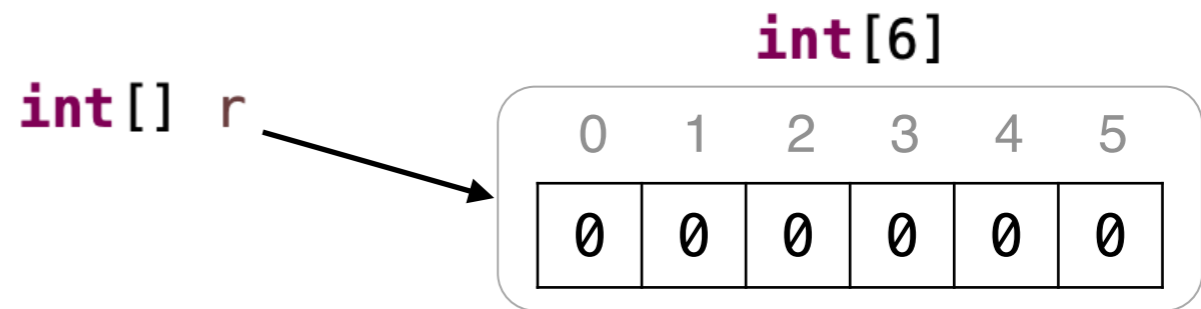
```
int[] r;
```

```
int[] r → null
```

```
int[] r;
```



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

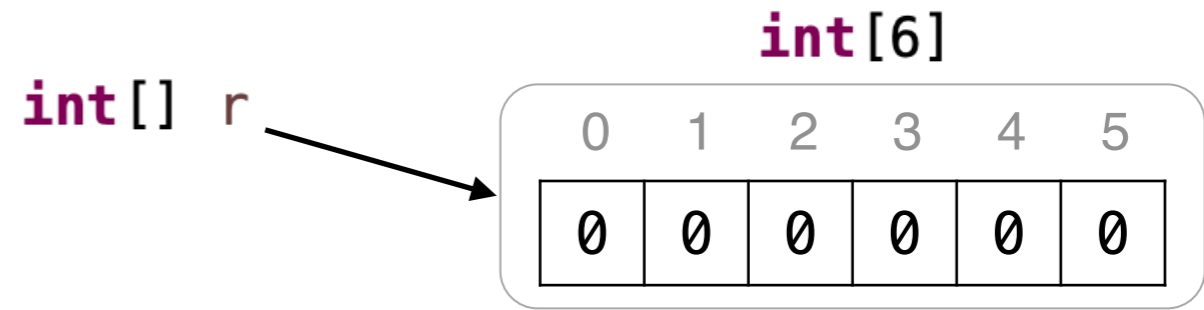


- 배열array은 객체object
- 배열 생성시 각 원소는 기본값으로 초기화

타입	초기값
byte char short int long	0
float double	0.0
boolean	false
Object	null

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

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



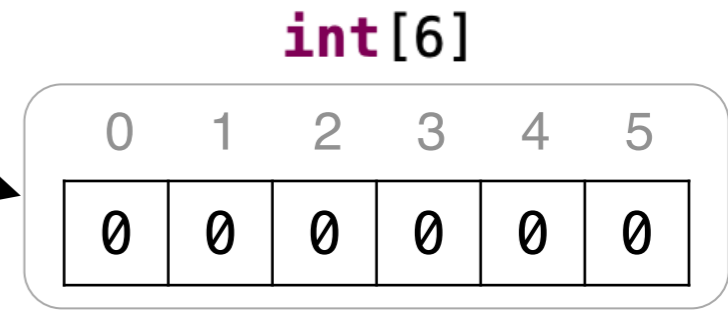
```
r[1] = 7;
```

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

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

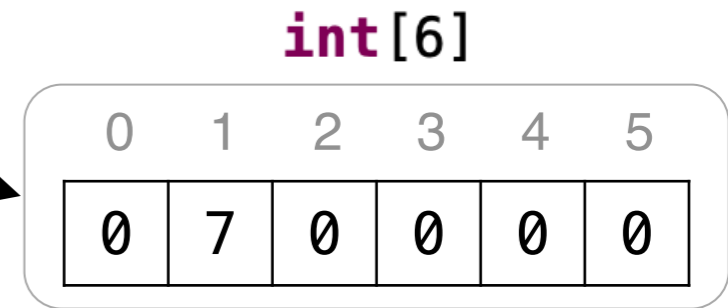
```
int[] r = new int[6];
```

```
int[] r
```



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

```
int[] r
```

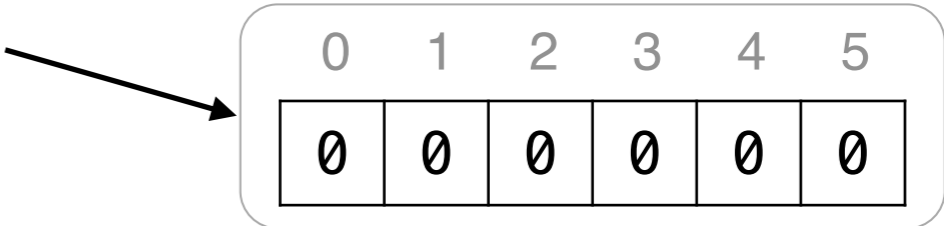


```
r[3] = r[1] + 2;
```

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

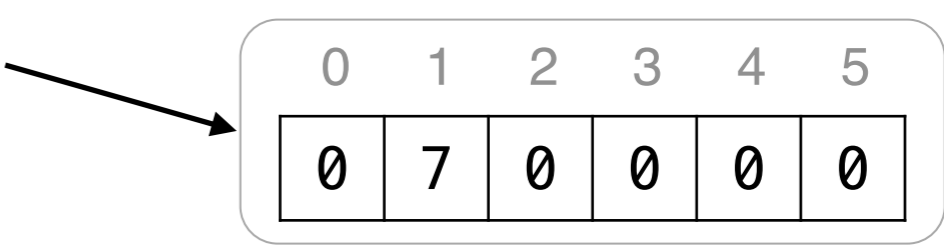
```
int[] r = new int[6];
```

```
int[] r
```



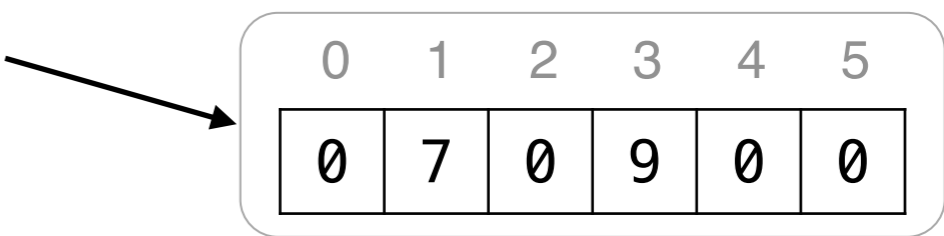
```
r[1] = 7;
```

```
int[] r
```



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

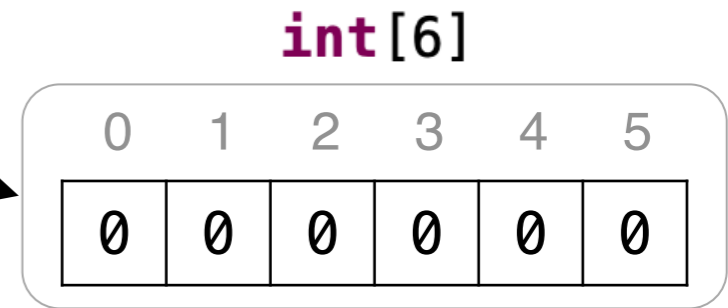
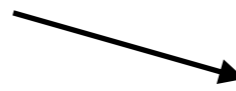
```
int[] r
```



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

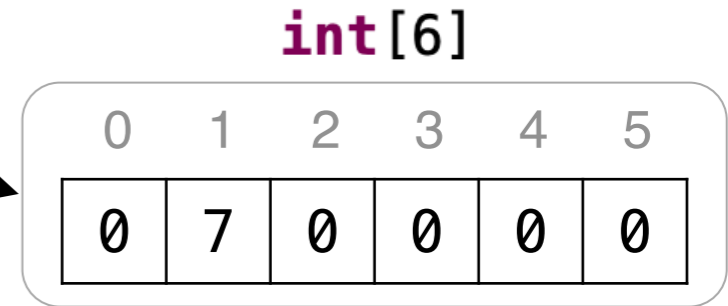
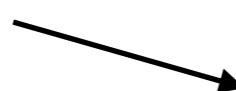
```
int[] r = new int[6];
```

```
int[] r
```



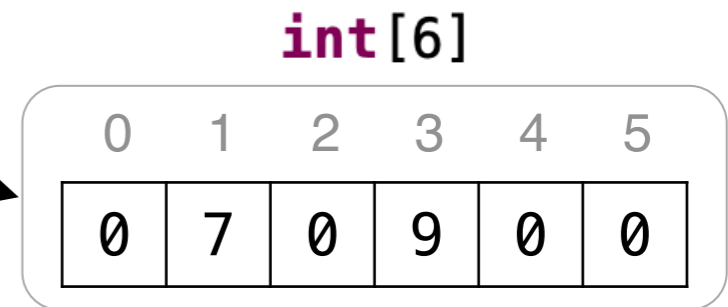
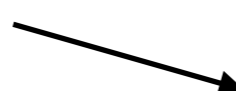
```
r[1] = 7;
```

```
int[] r
```



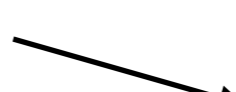
```
r[3] = r[1] + 2;
```

```
int[] r
```

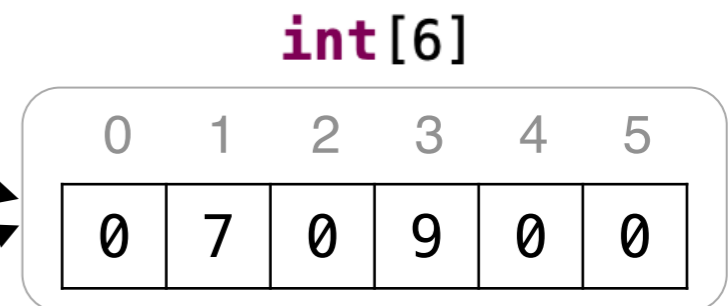


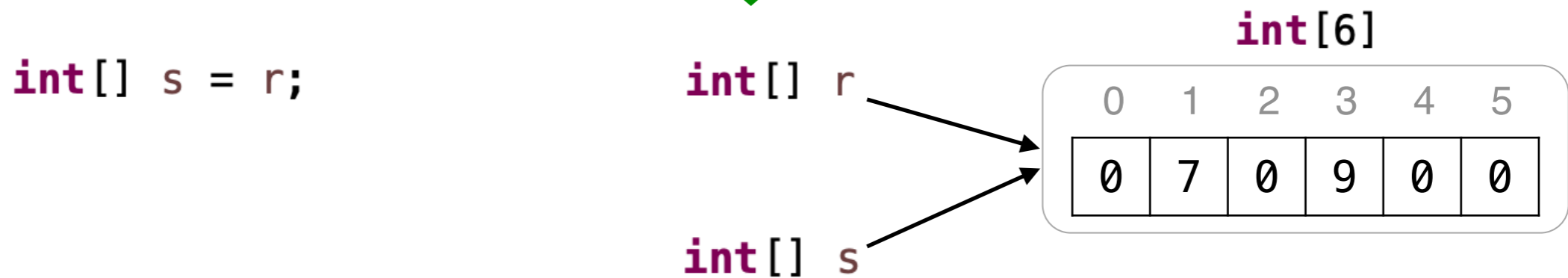
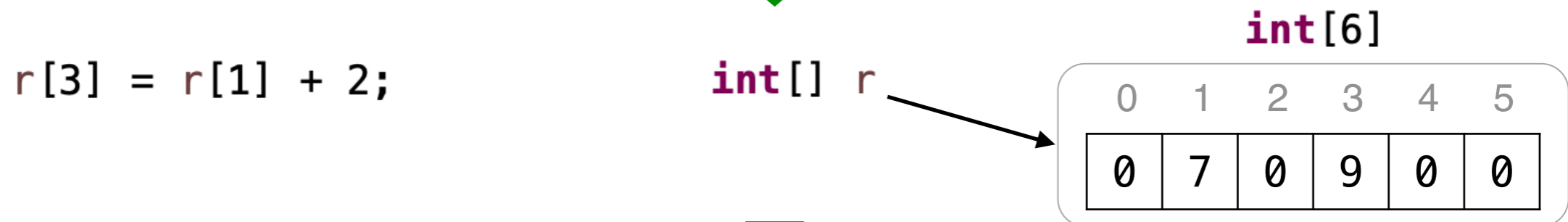
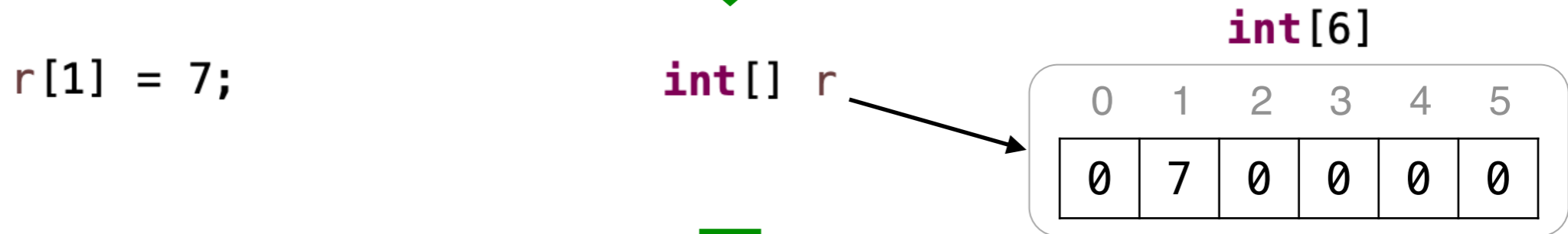
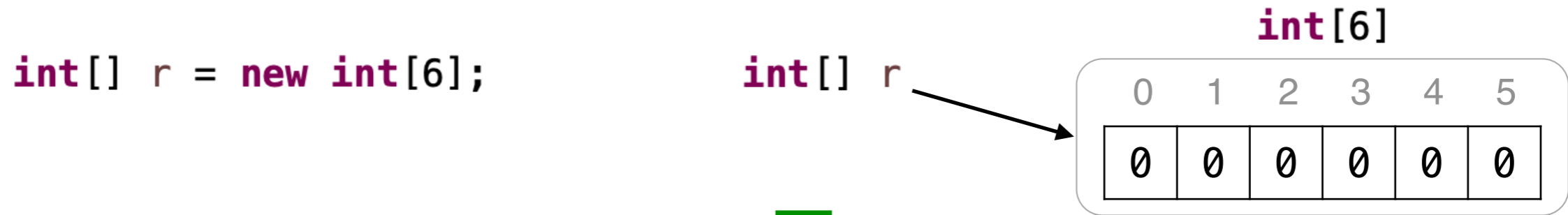
```
int[] s = r;
```

```
int[] r
```



```
int[] s
```





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

배열 생성 + 값 채우기

```
int[] r = new int[12];
```

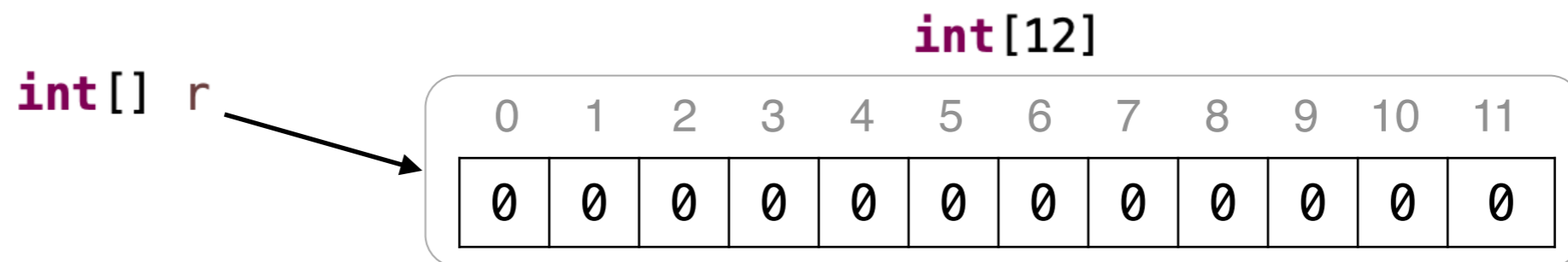
```
r[0] = 1;
```

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

```
for (int i = 2; i < r.length; i++) {
```

```
    r[i] = r[i-1] + r[i-2];
```

```
}
```



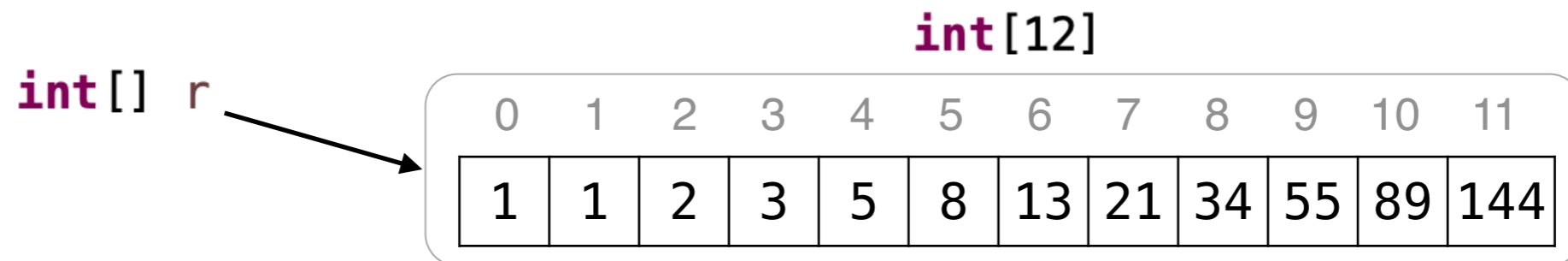
배열 생성 + 값 채우기

```
int[] r = new int[12];
```

```
r[0] = 1;
```

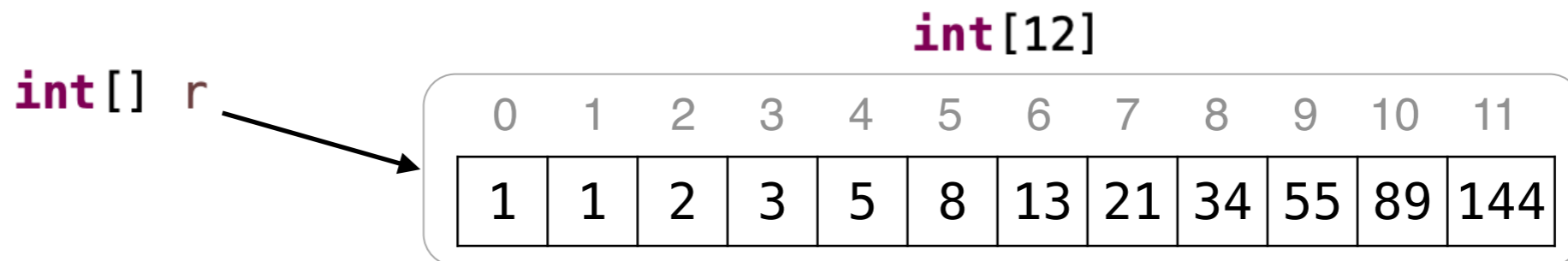
```
r[1] = 1;
```

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



배열 생성 지름길

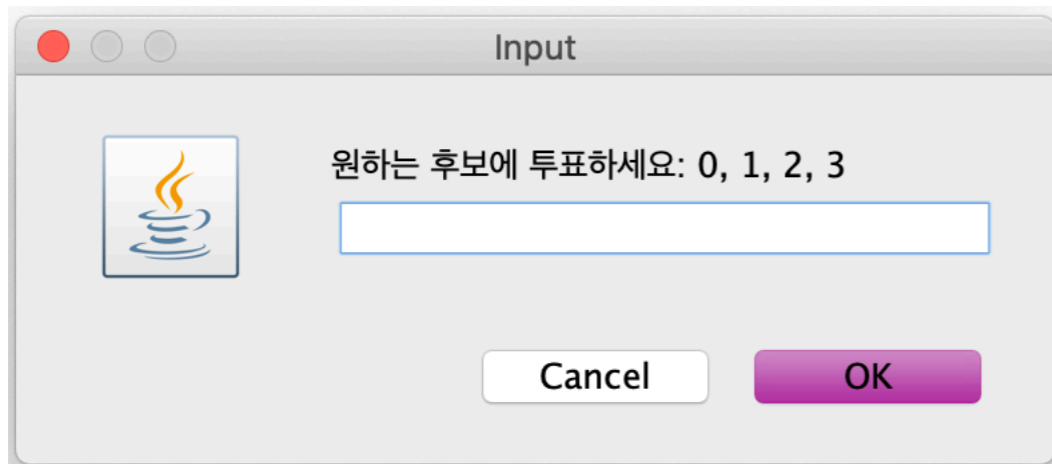
```
int[] r = {1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144};
```



사례 학습 1. 투표

입력

출력



A screenshot of a Java Swing dialog box titled "Input". The dialog box has a standard Mac OS window header with three colored buttons (red, yellow, green). On the left side, there is a Java logo icon. The main content area contains the text "원하는 후보에 투표하세요: 0, 1, 2, 3" followed by a text input field. At the bottom, there are two buttons: "Cancel" and "OK".

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

```

import javax.swing.*;

/** VoteCount 투표 애플리케이션 */
public class VoteCount {

    public static void main(String[] args) {
        int num_candidates = 4;
        int[] votes = new int[num_candidates]; // 0으로 자동 초기화

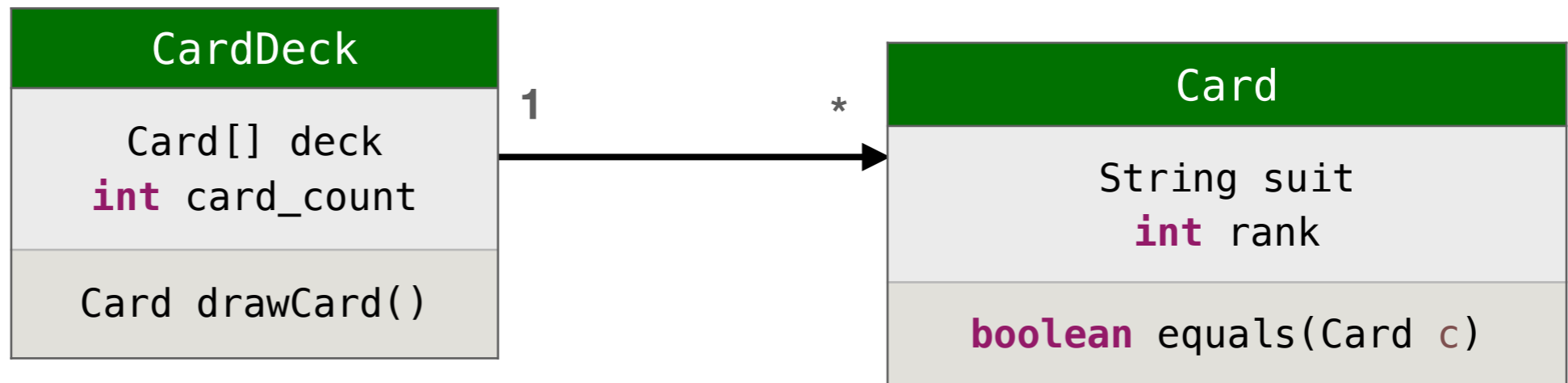
        boolean processing = true;
        while (processing) {
            /* loop invariant: 기표수는 해당 배열에 후보자별로 누적되었음 */
            String input = JOptionPane.showInputDialog("원하는 후보에 투표하세요: 0, 1, 2, 3");
            if (input == null)
                processing = false;
            else {
                char vote = input.charAt(0);
                if ('0' <= vote && vote <= '3') {
                    int voted = vote - '0';
                    votes[voted] += 1;
                }
                else
                    System.out.println(input + "은 후보자가 아닙니다.");
            }
        }
        for (int i = 0; i < votes.length; i = i + 1)
            System.out.println("후보 " + i + "번이 " + votes[i] + "표를 득표하였습니다.");
    }
}

```

사례 학습 2. 놀이 카드

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

Model



class	CardDeck	카드 1벌
field	Card[] deck	카드 덱
	int card_count	덱에 현재 남아있는 카드의 장수
method	Card drawCard()	덱에서 새 카드 한 장을 무작위로 뽑아줌, 없으면 카드 1벌을 새로 만들고 한 장을 무작위로 뽑아줌

class	Card	카드 1장
field	String suit	카드의 무늬
	int rank	카드의 등급
method	boolean equals (Card c)	카드 비교

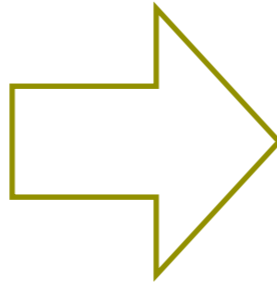
CardDeck, Card 클래스를 활용하여 다음 명세를 만족하는 Hand 클래스를 만들자.

class	Hand	카드 놀이 하는 참여자의 손에 들고 있는 카드
field	Card[] hand;	손에 들고 있는 카드를 담을 배열
	int number_of_cards	손에 들고 있는 카드의 장수
constructor	Hand(int max)	max는 들고 있을 수 있는 카드의 최대 장수
method	boolean receiveCard(Card c)	카드를 받아서 손에 추가하고 true를 리턴 한도를 초과하여 받을 수 없으면 false를 리턴
	boolean playCard(Card c)	카드가 있으면 손에서 제거하고 true를 리턴 없으면 false를 리턴
	void showHand()	들고 있는 카드를 모두 실행창에 보여 준다. 카드가 없으면 없음을 알려 준다.

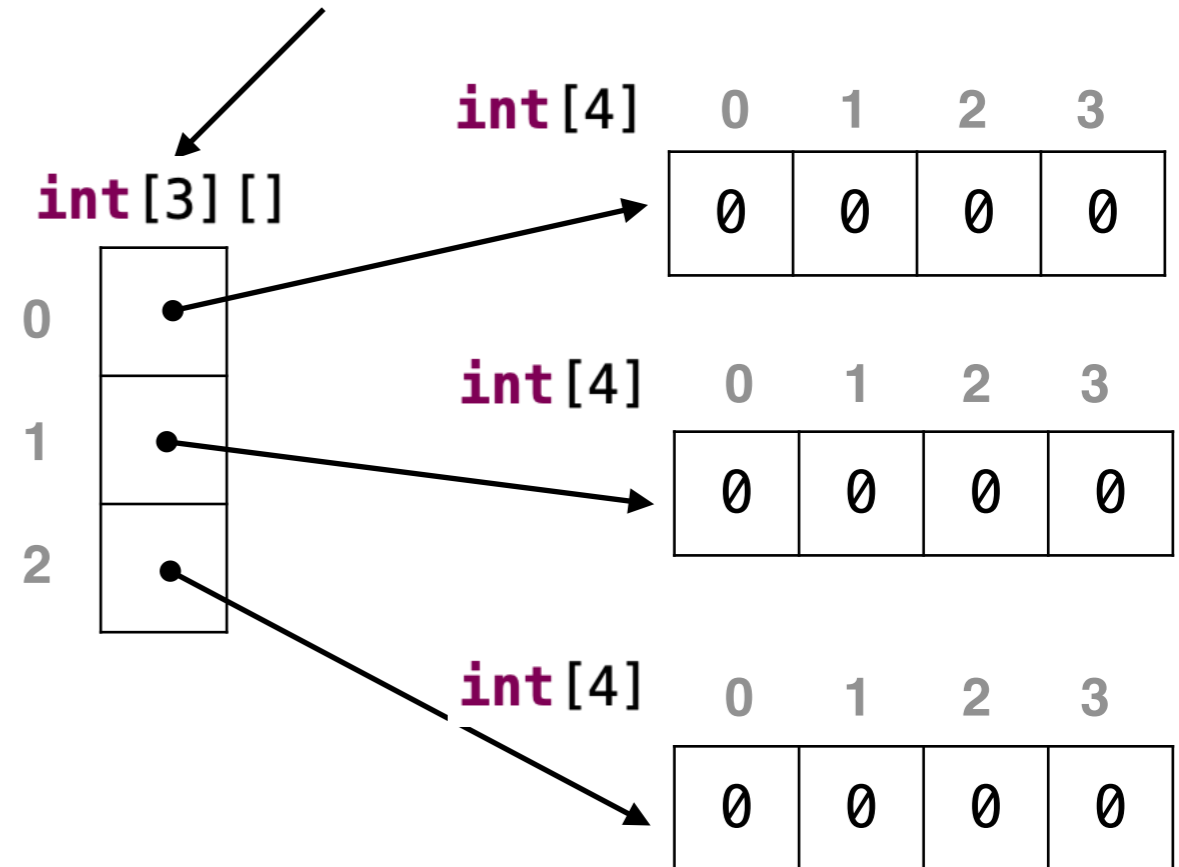
2차원 배열

사례 학습 3. 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`

2차원 배열

사례 학습 3. 학생회장 선거 득표 현황

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

```
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.");
```

```
}
```

1학년 0

2학년 1

3학년 2

Ragged Arrays 들쭉날쭉 배열

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

double[4] []

0	null
1	null
2	null
3	null

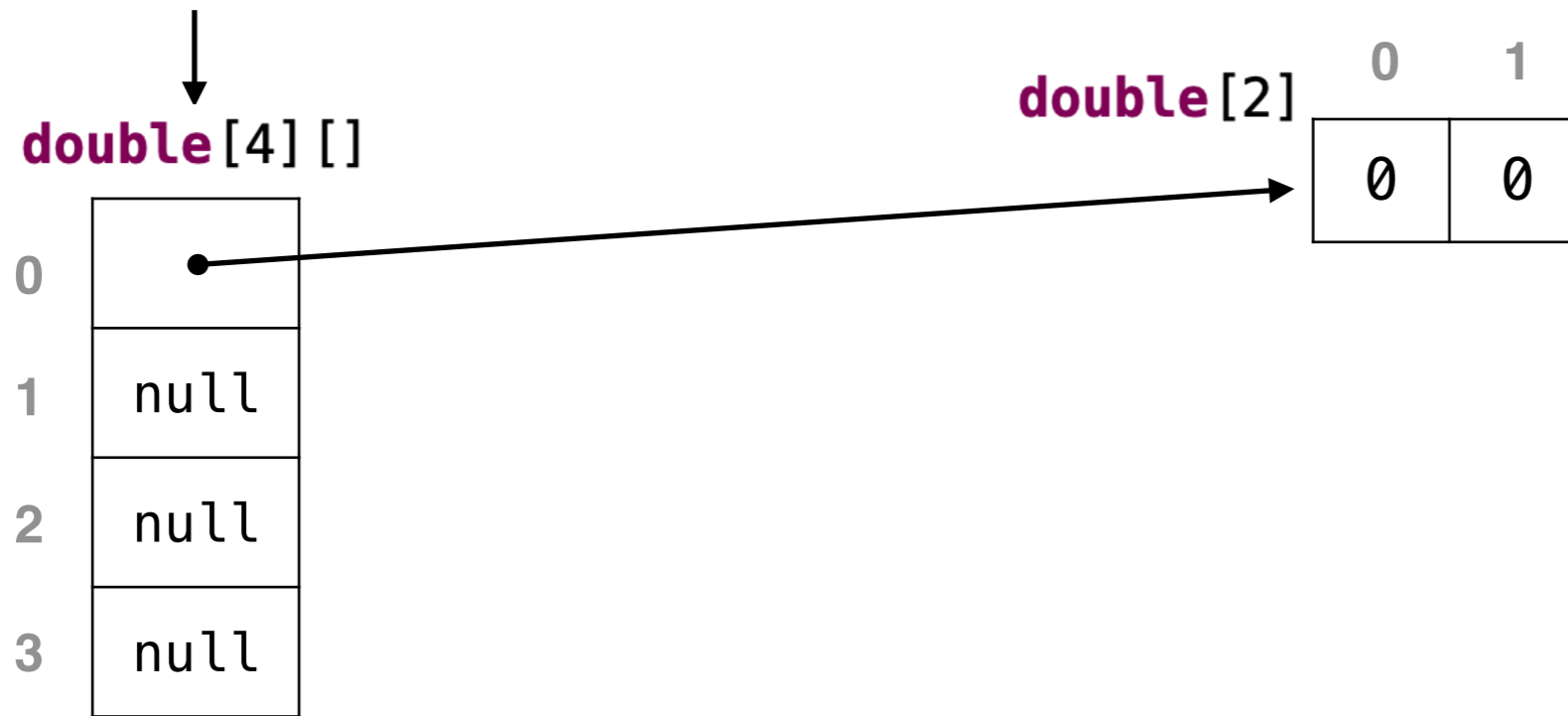
```
double[0] = new double[2];
```

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

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

Ragged Arrays 들쭉날쭉 배열

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



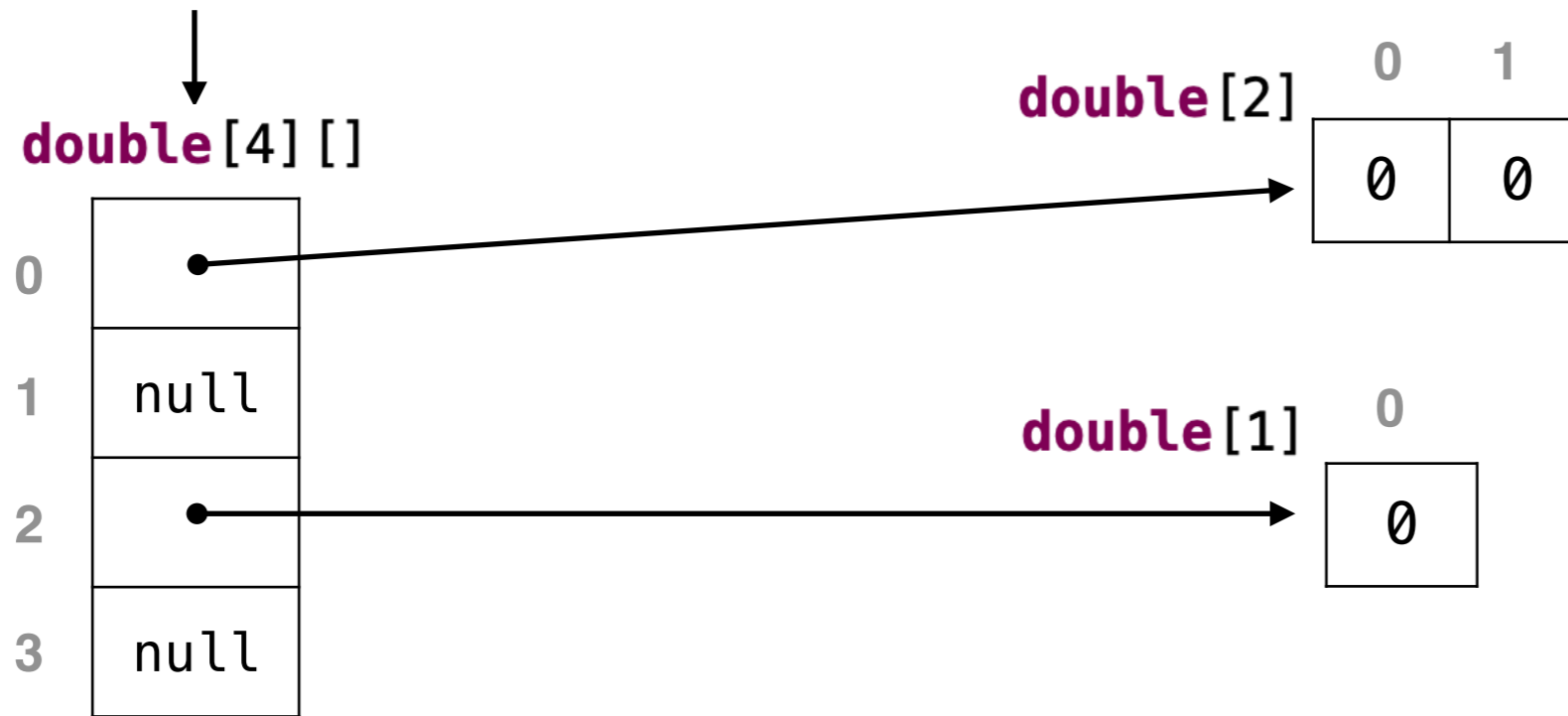
```
double [0] = new double [2];
```

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

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

Ragged Arrays 들쭉날쭉 배열

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



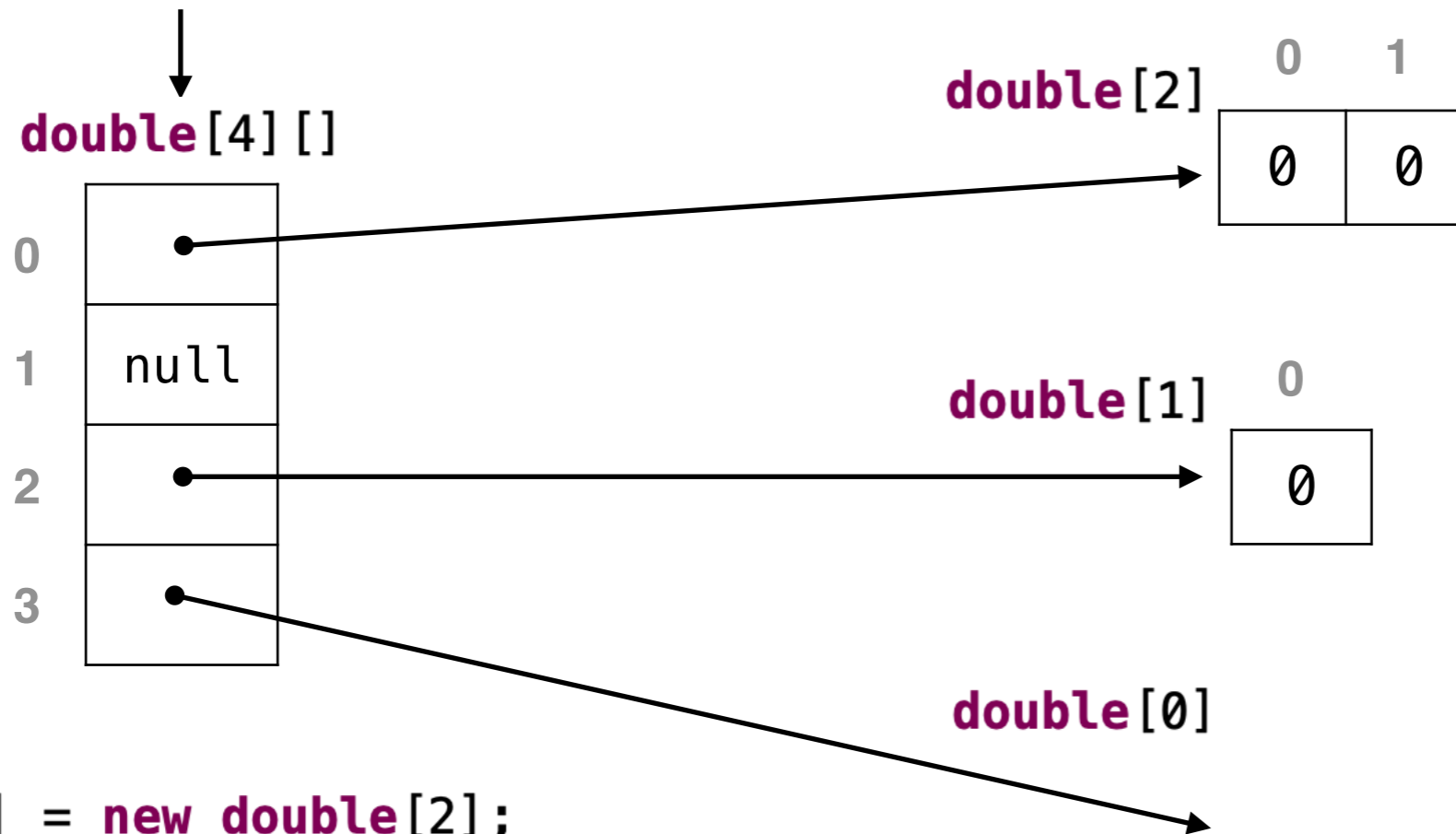
```
double [0] = new double [2];
```

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

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

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;
```

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

```
        else {
```

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

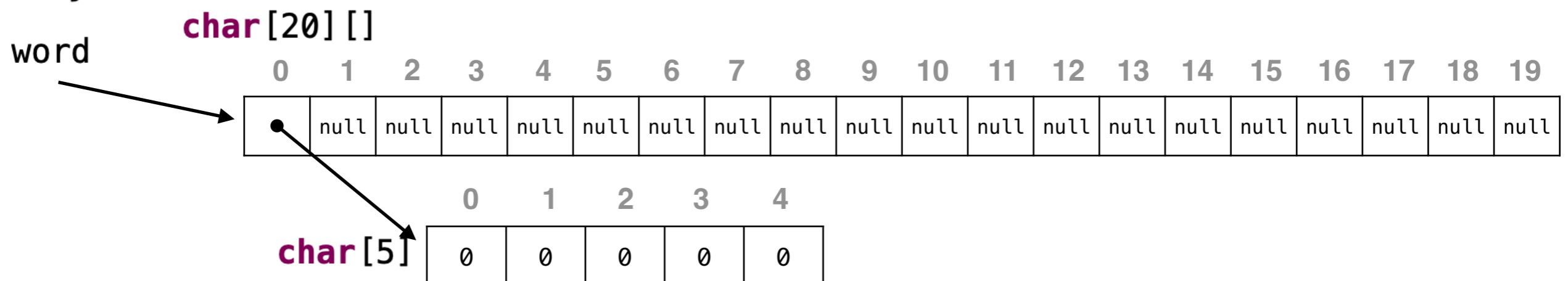
```
            for (int i = 0; i < s.length(); i++)
```

```
                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;
```

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

```
        else {
```

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

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

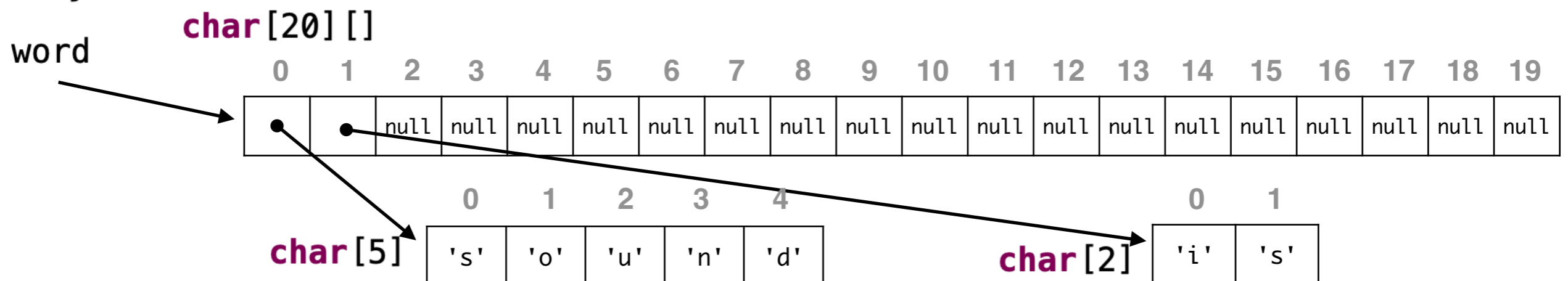
```
                word[count][i] = s.charAt(i);
```

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

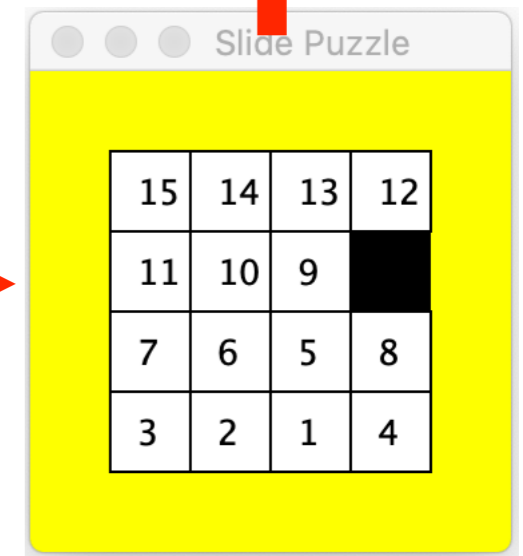
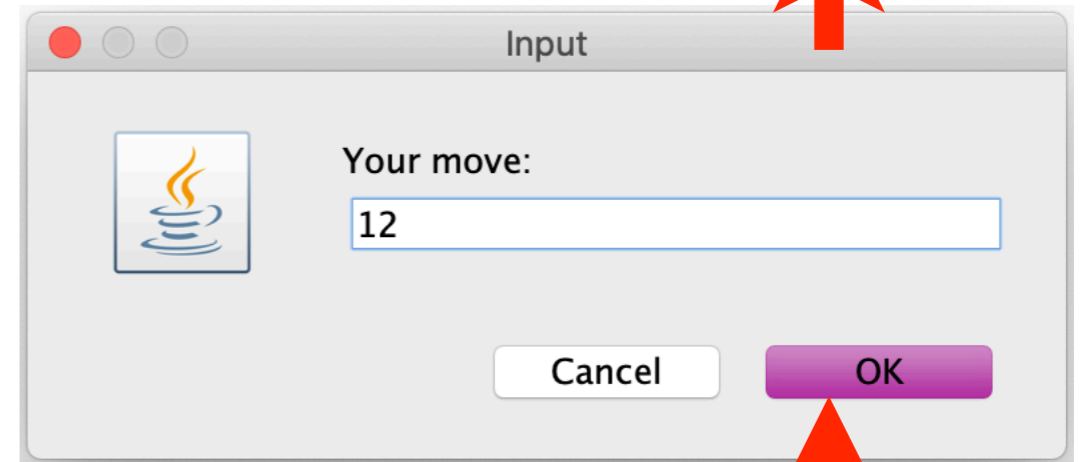
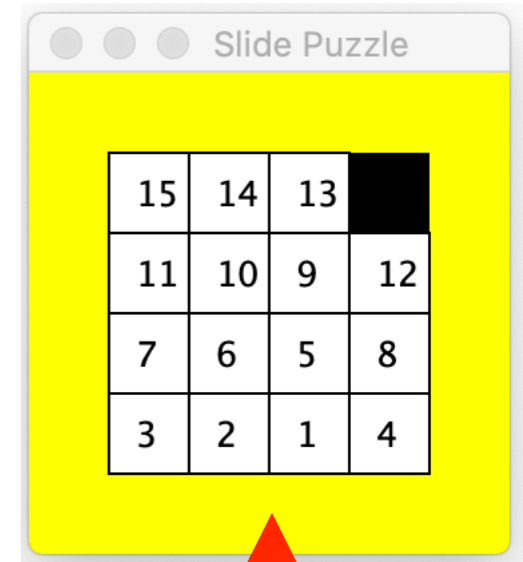
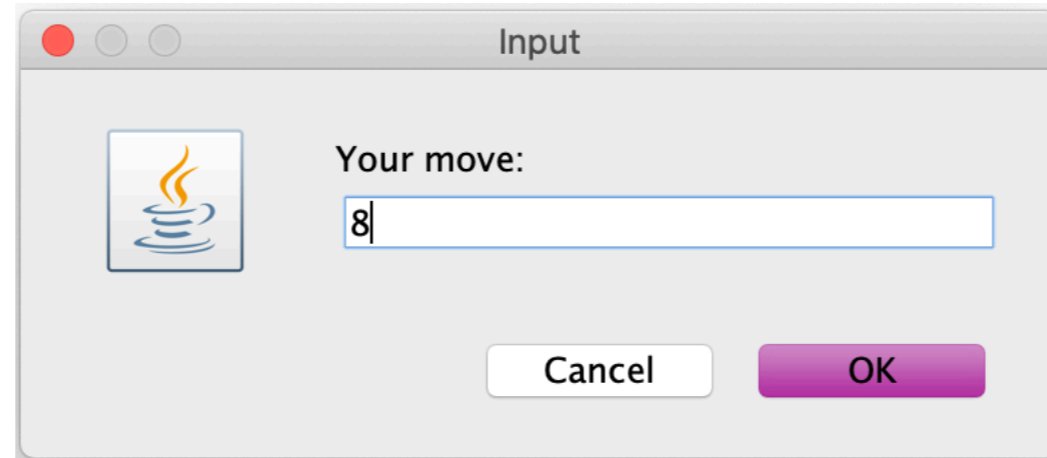
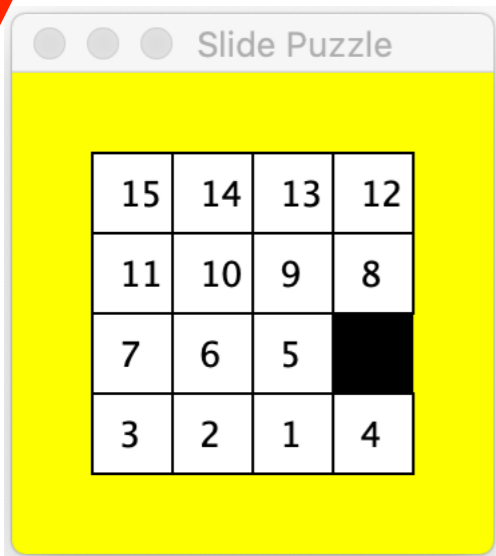
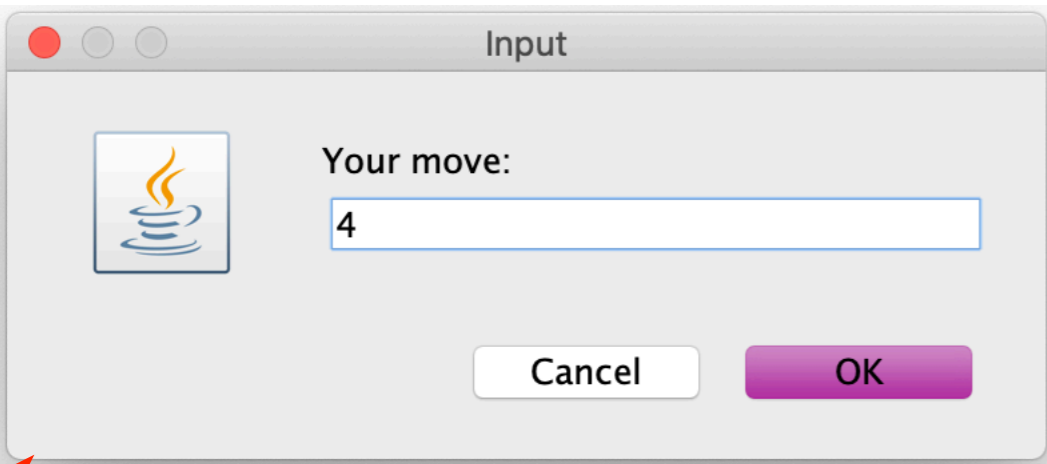
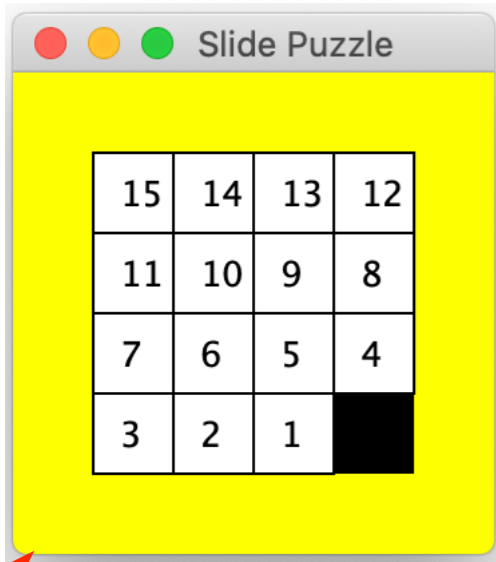
```
            count++;
```

```
        }
```

```
    }
```



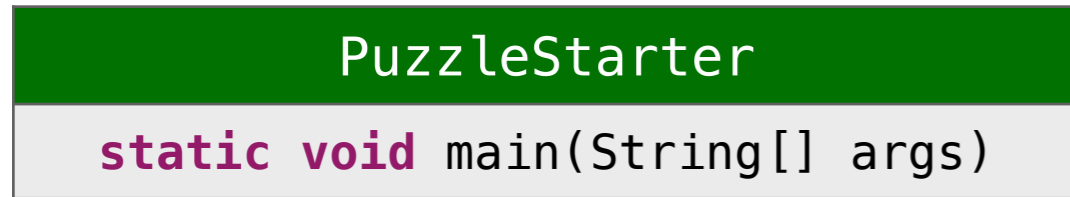
실습 - 슬라이드 퍼즐 게임



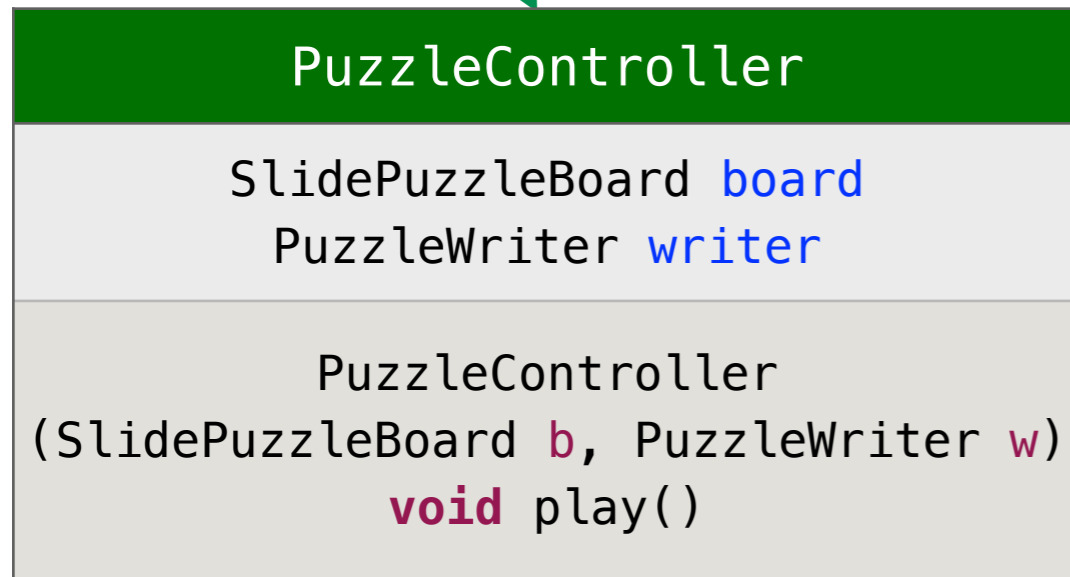
실습

슬라이드 퍼즐 게임

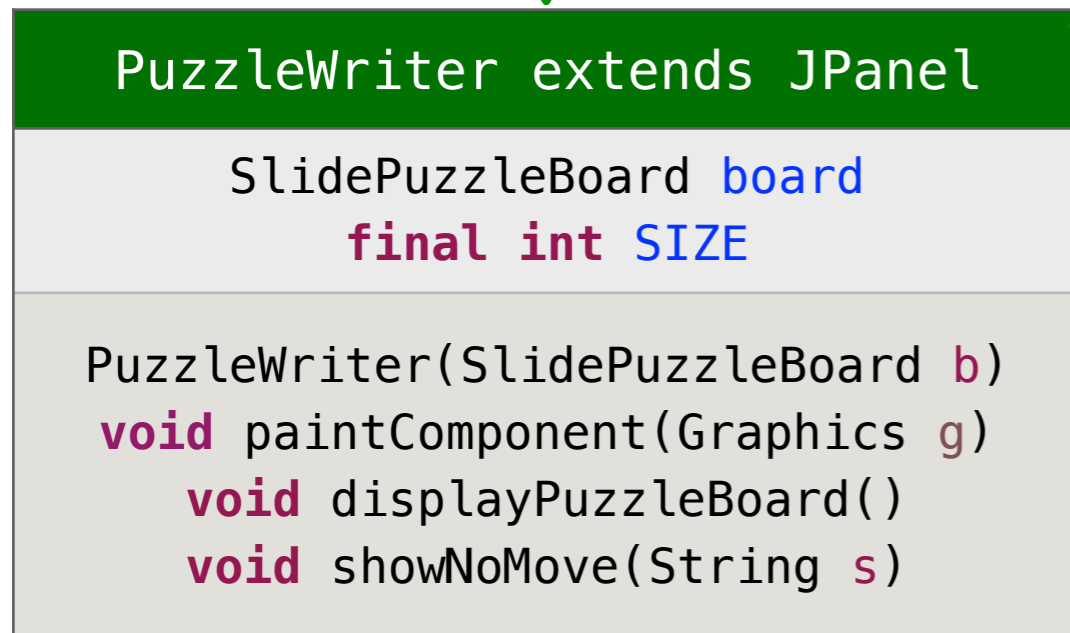
Starter



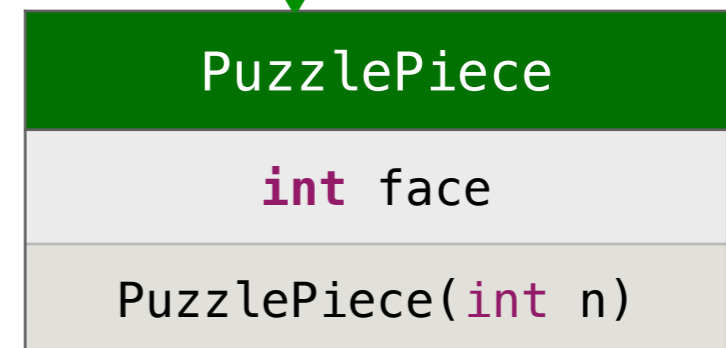
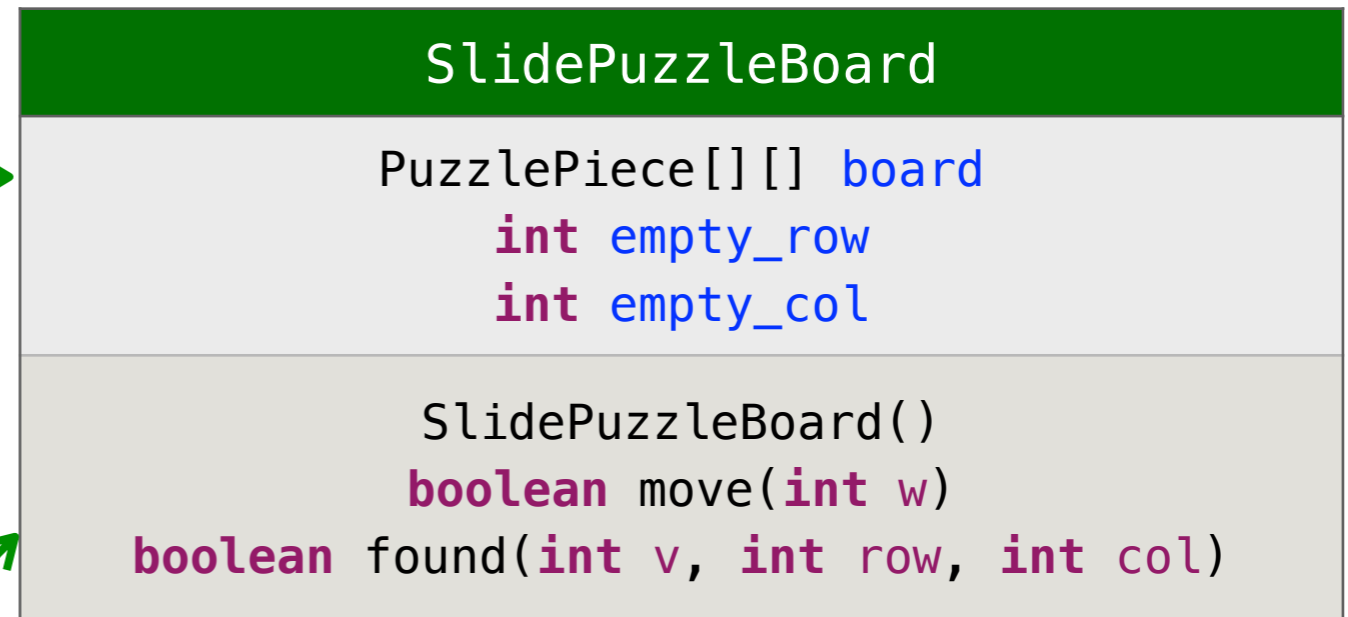
Controller



Output View



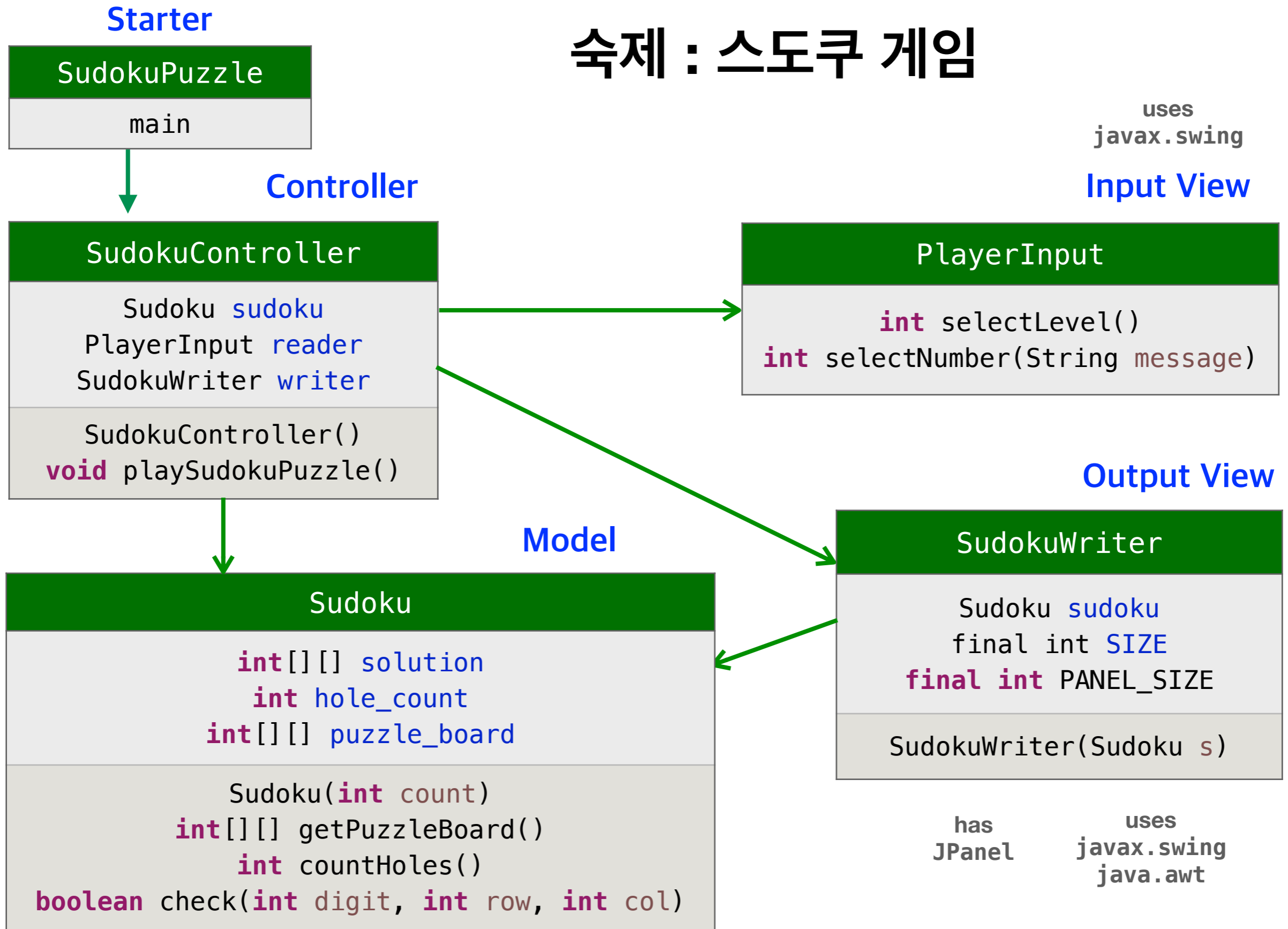
Model



has
JPanel
uses
javax.swing
java.awt

1
*

숙제 : 스도쿠 게임



숙제 : 스도쿠 게임

Sudoku

```
int[][] solution
int hole_count
int[][] puzzle_board
```

Sudoku(int count)

객체를 초기화 한다.
count - 빈칸의 개수

int[][] getPuzzleBoard()

퍼즐 보드 배열을 리턴 한다.

int countHoles()

빈칸의 개수를 리턴 한다.

boolean check(int digit, int row, int col)

row번 가로줄, col번 세로줄에 digit을 채울 수 있는지 검사하여, 가능하면 채우고 true를 리턴하고, 불가능하면 false를 리턴 한다.