

능동적 사고 방식의

java

강사 박주병

Park Ju Byeong

Park Ju Byeong



Part05 배열



01 배열의 선언과 생성

02 배열의 인덱스

03 다차원배열

04 실습 문제

메모리



CPU

코드 라인단위로
연산하여 데이터를
주고 받음



메모리

프로그램 실행



HDD,SSD





메모리주소	값
0x000A	10
0x000B	0x000D
0x000C	3.14
0x000D	홍
	길
	동



```
int score = 10;
```

```
String name = "홍길동";
```

```
final double PI = 3.14;
```



01 —

배열의 선언과 생성

학생 100명의 성적을 저장하려면?

```
int student1=10;  
int student2=30;  
int student3=100;  
int student4=50;  
int student5=35;  
int student6=70;
```

.....



int 변수100개;

배열

배열의 선언 방법

```
int[] student = new int[100];  
  
int student2[] = new int[100];
```

배열의 생성

```
int[] student = new int[100];  
  
int student2[] = new int[100];
```

선언과 생성 분리

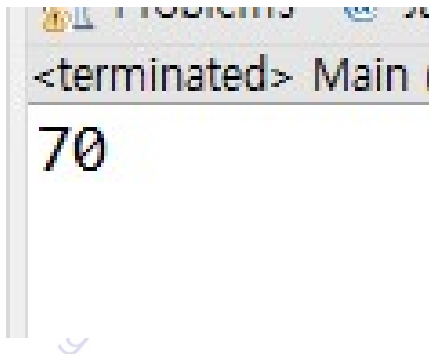
```
int[] student;  
  
student = new int[100];
```


배열 사용

```
int[] student= new int[5]; // 배열 생성
```

```
student[0] =50; //값 초기화  
student[1] =30;  
student[2] =70;
```

```
System.out.println(student[2]);
```



배열의 메모리 구조

- 같은 타입의 변수를 연속된 공간에 한번에 선언 하는것

```
int score = 10;
```

score



메모리주소	값
0x000A	10
0x000B	
0x000C	
0x000D	
0x000E	
0x000F	

```
int[] score = new int[100];
```

score



메모리주소	값
0x000A	0x000B
0x000B	score[0]
0x000C	score[1]
0x000D	score[2]
0x000E	score[3]
0x000F	score[4]

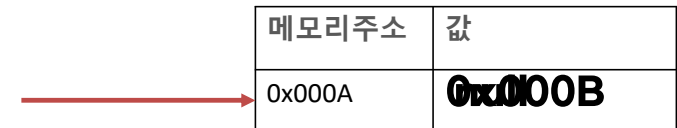
배열의 생성과정

```
int[] student;
```

```
new int[100];
```

```
student = new int[100];
```

student



메모리주소	값
0x000A	0x000B 0x000A

Null(널)

```
int[] student;
```

```
student[0] = 50;
```

왜 에러가 발생할까?

```
int[] student= new int[5];
```

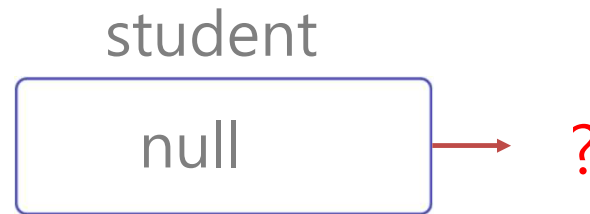
```
student[0] = 50;
```

메모리주소	값
0x000A	0
0x000B	0
0x000C	0
0x000D	0
0x000E	0
0x000F	0

메모리의 일정 영역을 할당한다.

```
int[] student;
```

```
student[0] = 50;
```



메모리주소	값
0x000A	0
0x000B	0
0x000C	0
0x000D	0
0x000E	0
0x000F	0

null은 데이터 없음을 의미한다.

주소를 저장하는 참조 타입 변수는 기본값이 null이다.

https://www.youtube.com/shorts/_dEqwTTaHMs

— 02

배열의 인덱스

배열 접근 방법

- 배열이름[인덱스]

```
int[] student = new int[5];  
student[0] = 100;  
student[2] = 50;
```

첫번째 요소(인덱스 0)

세번째 요소(인덱스 2)

메모리주소	값
0x000A	100
0x000B	0
0x000C	50
0x000D	0
0x000E	0

```
int a = student[2];
```

일반변수와 동일하게 사용

상속

- 배열 인덱스는 변수도 사용 가능하다.

```
int[] student= new int[100];
```

```
for(int i =0 ; i<100 ; i++)  
    student[i] = i;
```

```
for(int i =1 ; i<100 ; i++)  
    student[i-1] = i;
```

연산의 결과가 인덱스 범위에 속하면 된다.

배열의 범위

```
int[] student = new int[5];
```

```
student[0] = 0;
```

```
student[1] = 10;
```

```
student[2] = 20;
```

```
student[3] = 30;
```

```
student[4] = 40;
```

```
student[5] = 50;
```

→ 배열의 범위를 넘어서 ERROR!

길이 0 배열

```
int[] student = new int[0];
```

가능할까?

```
    student[0] = 30;  
}
```



The screenshot shows the console output of a Java application. The error message is: `Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 0 out of bounds for length 0`. The stack trace indicates the error occurred at `joo.강의5.Main.main(Main.java:16)`. The IDE interface includes tabs for Problems, Javadoc, Declaration, Console, Git Staging, Debug, and Call Hierarchy.

길이가 0이기에 사용은 할 수 없다 ERROR!

왜 쓰는걸까?

길이가 0 이기에 어차피 사용 할 수 없다.

`int [] student;` **VS** `int [] student = new int [0];`

배열의 length 변수

```
int[] student= new int[4];  
System.out.println(student.length);
```

Console
<terminated>
4

```
student.length = 10;
```

→ 상수라서 값을 변경 할 수 없다!

```
int[] student= new int[5];  
  
for(int i =0 ; i<5;i++)  
    System.out.println(student[i]);
```



```
int[] student= new int[5];  
  
for(int i =0 ; i<student.length;i++)  
    System.out.println(student[i]);
```

배열의 길이를 리터럴로 직접
적어주는것보다
length를 사용하는것이 좋다.

```
int [] student;      int [] student = new int [0];
```

```
for(int i=0 ;i<student.length;i++)  
    System.out.println(student[i]);
```

길이0의 배열을 활용하면 배열을 null 체크를 따로 하지 않아도 되며 에러가 발생하지 않는다.

배열의 요소 초기화

```
int a;  
  
System.out.println(a);
```

→ 일반 변수는 초기화 하지 않으면 사용할 수 없다

```
int[] student= new int[3];  
  
for(int i =0 ; i<student.length;i++)  
    System.out.println(student[i]);
```

→ 해당 타입에 맞게 자동 초기화 되어 있다.

```
int[] student= new int[] {30,20,10};
```

→ 직접 초기화 값을 지정할 수 있다.

```
int[] student= new int[3] {30,20,10};
```

→ 초기화값을 적어주었다면 크기 지정은 불가하다 ERROR!

```
int[] student= {30,20,10};
```

→ 배열의 타입을 선언부에서 알 수 있기때문에 생략해도 된다.

```
int[] student;  
  
student = {30,20,10};
```

→ 선언과 초기화 분리시 new int[] 를 생략할 수 없다. ERROR!

```
int[] student= new int[]{};
```

→ 길이 0의 배열이 생성된다.

```
int[] student= {};
```

→ 길이 0의 배열이 생성된다.

배열 요소의 출력

```
for(int i =0 ; i<student.length;i++)  
    System.out.println(student[i]);
```

반복문으로 하나씩 출력 가능하다.

크기 5의 배열을 만들고 사용하던 중 더 크게 필요하다면 ?(심화)

```
int[] arr= new int[5];  
  
int[] temp = new int[arr.length*2];  
  
for(int i =0 ; i<arr.length;i++)  
    temp[i]=arr[i];  
  
arr = temp;
```

arr[5]
0x000A

temp[10]

0x300A

주소복사!

메모리주소	값
0x000A	30
0x000B	40
0x000C	50
0x000D	60
0x000E	70

메모리주소	값
0x300A	0
0x300B	0
0x300C	0
0x300D	0
0x300E	0
0x300F	0
0x3010	0
0x3011	0
0x3012	0
0x3013	0

값복사

향상된 for문(foreach)

```
int[] arr = new int[10];  
  
for(int i = 0 ; i < arr.length; i++)  
    arr[i] = (int)(Math.random()*10);
```

기존의 For문을 이용한 배열 순차 접근



```
for(int i : arr)  
    System.out.println(i);
```

배열을 순차접근하는 목적이라면 향상된 for문을 사용하자!

1-1 실습문제 (normal)

int 타입 길이5의 배열을 만들어 임의의 값을 넣고 출력해보자
- 출력은 반복문을 이용해 보자

```
0  
10  
20  
30  
40
```

1-1 문제풀이 (normal)

Park Ju-seong

Park Ju Byeong

1-2 실습문제 (normal)

1번 문제에서 만들어놓은 배열 요소들의 평균을 구해보자

```
int[] arr = {10,20,30,40,50};
```

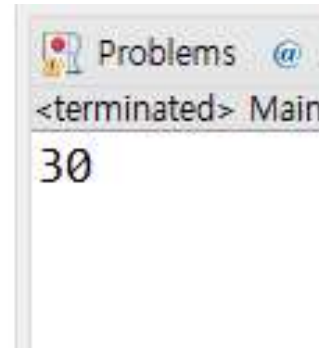
```
arr[0] = 10;
```

```
arr[1] = 20;
```

```
arr[2] = 30;
```

```
arr[3] = 40;
```

```
arr[4] = 50;
```



1-2 문제풀이 (normal)

1-3 실습문제 (normal)

int 타입 길이 10의 배열을 만들어 임의의값을 넣고 최대값과 최소값을 출력하자

```
int[] arr = {5,478,41,2,45,758,14,3,97,10};
```

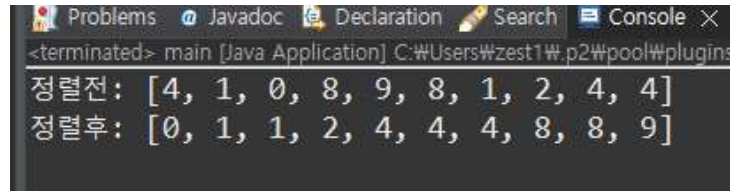
Problems @ Javadoc
<terminated> Main (6) [Java
최대값은 758
최소값은 2

1-3 문제풀이 (normal)

1-4 실습문제 (hard)

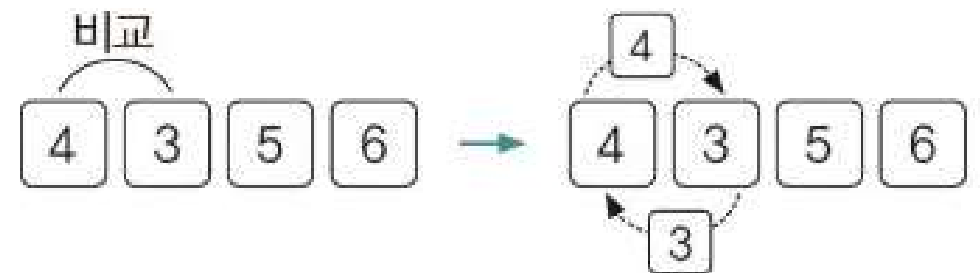
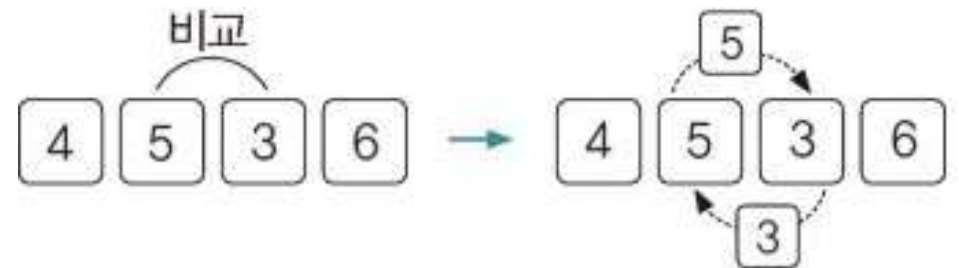
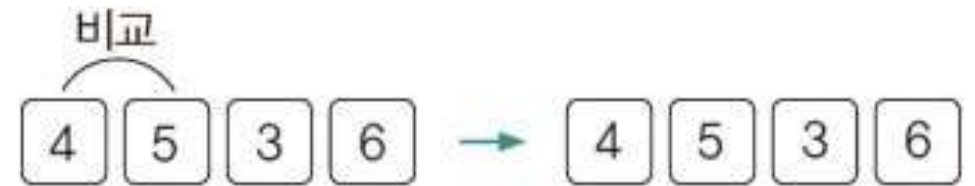
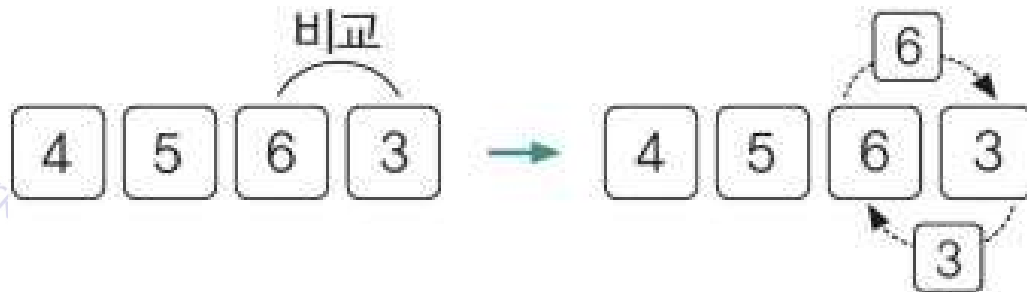
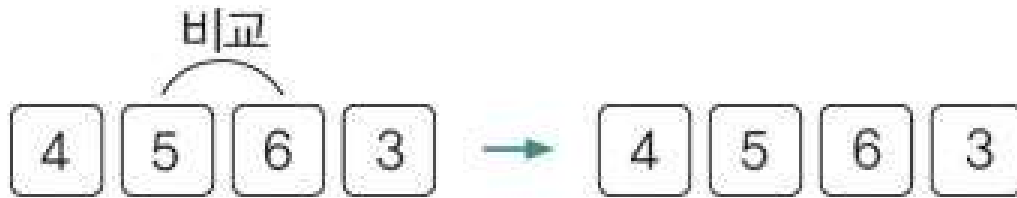
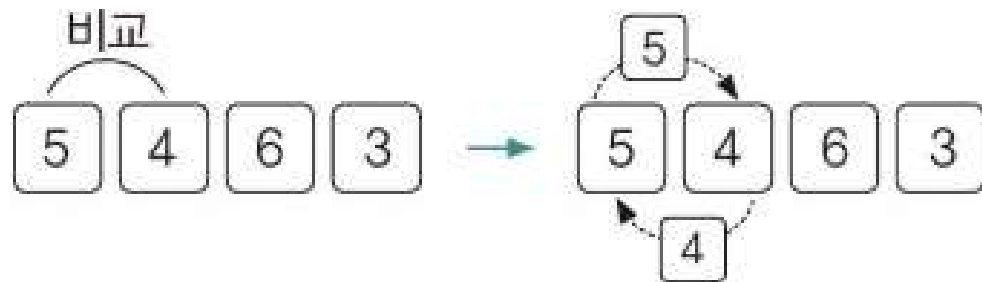
int타입 길이10의 배열에 0~9 까지의 랜덤한값을 넣은후
오름차순으로 정렬하여보자

```
int[] arr = new int[10];  
  
for(int i = 0 ;i<arr.length; i++)  
    arr[i] = (int)(Math.random()*10);  
  
System.out.println("정렬전: "+ Arrays.toString(arr));
```

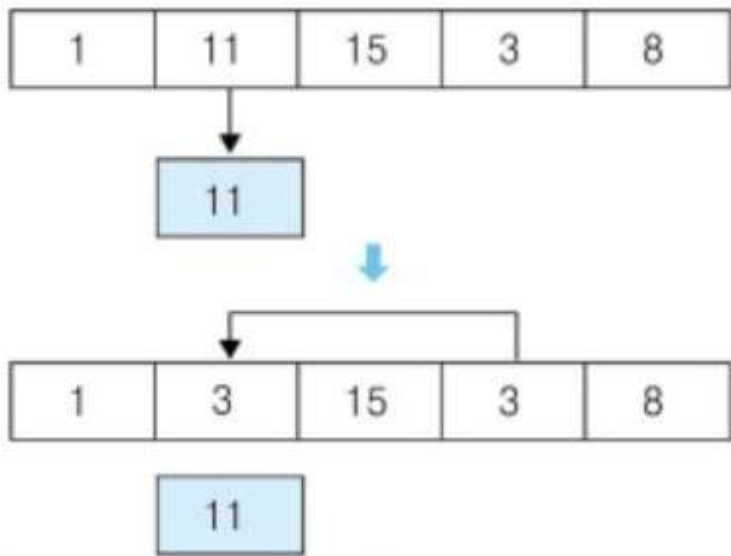
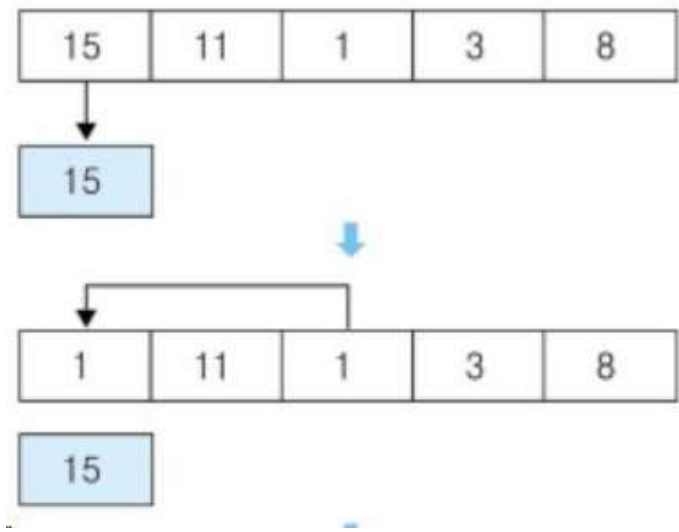


Problems Javadoc Declaration Search Console X
<terminated> main [Java Application] C:\Users\Wzest1\p2\pool\plugins
정렬전: [4, 1, 0, 8, 9, 8, 1, 2, 4, 4]
정렬후: [0, 1, 1, 2, 4, 4, 4, 8, 8, 9]

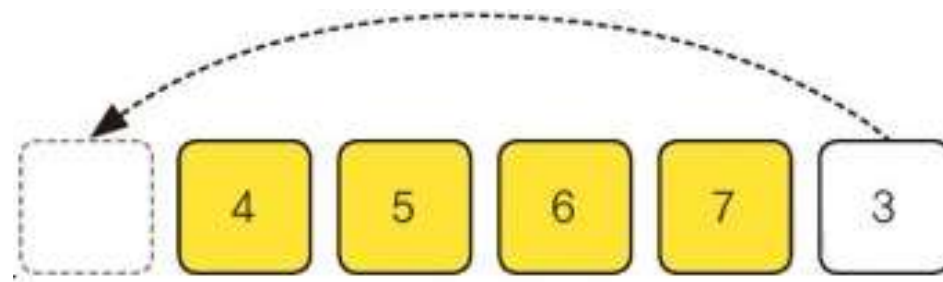
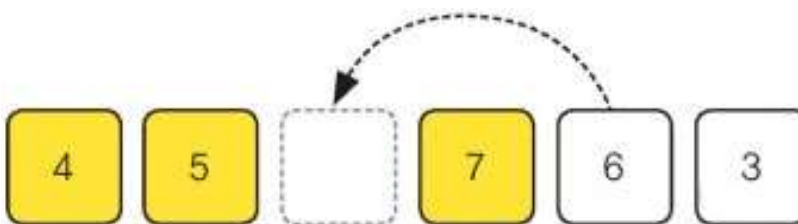
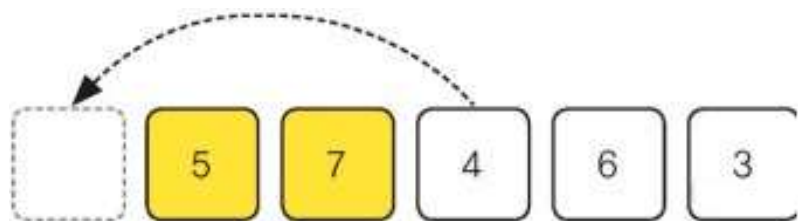
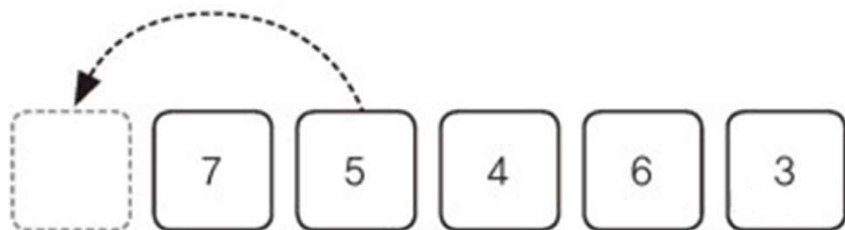
버블 정렬



선택 정렬



삽입 정렬





03

다차원배열

2차원 배열의 선언과 생성

```
int[][] arr = new int[5][3];
```

3열

5행

arr[0]	int	int	int
arr[1]	int	int	int
arr[2]	int	int	int
arr[3]	int	int	int
arr[4]	int	int	int

길이3의 배열을 저장하는 길이5배열

2차원 배열의 index

int	int	int
int	int	int
int	int	int
int	int	int
int	int	int

arr[2][1]

2차원 배열의 초기화

```
int[][] arr = new int [][]{{0,1,2},{0,1,2}};
```

```
int[][] arr = {{0,1,2},{0,1,2}};
```

```
int[][] arr = {  
    {0,1,2}  
    ,{0,1,2}  
};
```

int	int	int
int	int	int

2차원 배열 예시

```
int[][] arr = new int[5][3];
```

```
arr[0][0] = 10;  
arr[0][1] = 20;  
arr[0][2] = 30;
```

```
arr[1][0] = 40;  
arr[1][1] = 50;  
arr[1][2] = 60;
```

10	20	30
40	50	60
0	0	0
0	0	0
0	0	0

2차원 배열의 length는 얼마일까?

```
int[][] arr = new int [5][3];  
System.out.println(arr.length);
```



```
<terminated>  
5
```

int	int	int
int	int	int
int	int	int
int	int	int
int	int	int

```
int[][] arr = new int [5][3];  
System.out.println(arr[3].length);
```



```
<terminate  
3
```

int	int	int
int	int	int
int	int	int
int	int	int
int	int	int

for문을 이용한 2차원배열 초기화

```
int[][] arr = new int [5][3];

for(int i =0; i<arr.length;i++)
    for(int j =0; j<arr[0].length;j++)
        arr[i][j] = 50;
```

```
int[][] arr = new int [5][3];

for(int a : arr)
{
    System.out.println(arr[a]);
}
```


arr의 요소들은
길이3의배열이다. ERROR!

```
int[][] arr = new int [5][3];

for(int[] temp : arr)
    for(int i : temp)
        System.out.println(temp[i]);
```

가변배열

```
int[][] score = new int [5][3];
```




int	int	int
int	int	int
int	int	int
int	int	int
int	int	int

```
int[][] score = new int [5][];
```

```
System.out.println(score[0]);
```

```
null
```



Null
Null
Null
Null
Null

```
int[][] score = new int [5][];
```

```
score[0] = new int[3];  
score[1] = new int[3];  
score[2] = new int[3];  
score[3] = new int[3];  
score[4] = new int[3];
```

가변배열

```
int[][] score = new int [5][];  
  
score[0] = new int[4];  
score[1] = new int[3];  
score[2] = new int[2];  
score[3] = new int[1];  
score[4] = new int[6];
```

```
int[][] score = {{30,10,20}  
                 ,{20,10}  
                 ,{10,40,61,32,42}  
                 ,{45}  
                 };
```

int	Int	Int	Int		
int	Int	Int			
int	Int				
int					
int	Int	Int	Int	Int	Int

2차원배열의 각 요소인 1차원배열의 길이는 달라도 된다!

2-1 실습문제 (normal)

학생 5명의 국어, 영어, 수학 성적을 저장하는 2차원 배열을 만들어 저장하고 각 학생의 평균을 출력하여 보자(길이3의 1차원배열 5개를 저장 해야 한다)

- 데이터는 임의로 넣어보자
- 평균도 배열에 포함시켜도 된다.
- 간격을 일정하게 만들고 싶으면 `wt` 를 문자열에 포함하면 된다.

```
int[][] score = new int [][] {  
    {30, 50, 30}  
    , {70, 20, 90}  
    , {100, 80, 70}  
    , {90, 40, 30}  
    , {10, 40, 100}  
}  
;  
System.out.println("국어 \t영어 \t수학 \t평균");
```

```
<terminated> Main (11) [Java Application] C:\Users\USER545\p2\w  
국어      영어      수학      평균  
30         50         30         36  
70         20         90         60  
100        80         70         83  
90         40         30         53  
10         40         100        50
```

2-1 문제풀이 (normal)

2-2 실습문제 (hard)

차원배열을 이용하여 숫자를 입력받아 암호화 하는 프로그램을 만들자.

```
char[][] encryptGrid = {{'0',' '},
                        {'1','!'},
                        {'2','@'},
                        {'3','#'},
                        {'4','$'},
                        {'5','%'},
                        {'6','^'},
                        {'7','&'},
                        {'8','*'},
                        {'9','('}};
```

```
Scanner scan = new Scanner(System.in);
System.out.println("암호화할 숫자를 입력하세요 : ");
String input = scan.nextLine();
//사용자로 부터 받은 숫자
char[] plainText = input.toCharArray();
//암호화된 숫자를 저장하기 위한 배열
```

- nextLine() 은 키보드로부터 문자열을 입력받는다.
- input.toCharArray()는 문자열을 char 배열로 변환해준다.

```
<terminated> main [Java Application] C:\Use
암호화할 숫자를 입력하세요 :
45674
평문 : 45674
암호화 : $%^&$
```

2-2 문제풀이 (hard)

2-3 실습문제 (expert)

1~25까지 숫자를 순서대로 저장하는 2차원 배열[5][5]을 만든후 값을 랜덤하게 섞어보자

- 섞는방법: Math.random() 이용하여 1~25 사이 숫자를 뽑은후 2차원배열에서 동일한 숫자를 찾아 [0][0] 의 값과 교체하는것을 100번 반복하자)

```
int[][] arr = new int[5][5];

System.out.println("---서플전---");
for(int i=0;i<arr.length;i++)
{
    for(int j =0;j<arr[0].length;j++)
    {
        arr[i][j] = i*arr[0].length+(j+1);
        System.out.print(arr[i][j] + "\t");
    }
    System.out.println();
}

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

```
//숫자 셔플
for(int shuffleCnt=0;shuffleCnt<100;shuffleCnt++)
{
    int randomNumber = (int)(Math.random()*25)+1;

    for(int i=0;i<arr.length;i++)
    {
        for(int j =0;j<arr[0].length;j++)
        {
            //랜덤수와 동일한 숫자를 찾으면
            if(arr[i][j] == randomNumber)
            {
                //0,0 의 숫자와 바꿔서 섞는다.
                int temp = arr[i][j];
                arr[i][j] = arr[0][0];
                arr[0][0] = temp;
                break;
            }
        }
    }
}
```

```
---서플전---
1      2      3      4      5
6      7      8      9      10
11     12     13     14     15
16     17     18     19     20
21     22     23     24     25
-----
---서플후---
10     21     3      13     2
11     7      1      19     14
18     4      20     12     5
8      23     16     9      6
25     22     17     24     15
-----
```

2-3 문제풀이 (expert)

— 04

실습문제

3-1 실습문제 (hard)

좌표를 입력 받아 해당좌표에 x를 표시하자

```
char[][] arr = new char[][]{
    { ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ' },
    { ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ' },
    { ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ' },
    { ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ' },
    { ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ' },
};

Scanner scan = new Scanner(System.in);

System.out.println("좌표를 입력하세요:");
int pos = scan.nextInt();
```

좌표를 입력하세요:


```
24
 12345
1
2  X
3
4
5
```


3-1 문제풀이 (hard)

3-2 실습문제 (expert)

아래의 그림은 2차원 배열을 이용하여  모양의 테트리스 블록을 표현한 것이다. 블록을 시계방향으로 90도 돌려서  모양을 만들어 보자.

```
char[][] star = {  
    {'*', '*', ' ', ' ', ' ', ' '},  
    {'*', '*', ' ', ' ', ' ', ' '},  
    {'*', '*', '*', '*', '*', '*'},  
    {'*', '*', ' ', ' ', ' ', ' '},  
    {'*', '*', ' ', ' ', ' ', ' '},  
};  
  
char[][] result = new char[star[0].length][star.length];  
for(int i=0; i < star.length; i++)  
{  
    for(int j=0; j < star[i].length; j++)  
        System.out.print(star[i][j]);  
  
    System.out.println();  
}
```

- 2차원 배열[5][5]를 만든후 적절한 인덱스에 *을 넣어서  모양을 만든다.
- *의 위치를 다른 인덱스로 옮겨서 90도 회전을 시켜보자.

```
**  
  
**  
  
*****  
  
**  
  
**  
  
  
*****  
*****  
  
*  
  
*  
  
*
```

3-2 문제풀이 (expert)

3-3 실습문제 (expert)

1~25 사이의 셔플된 2차원 배열[5][5]을 만든후 사용자로부터 좌표를 입력받아 해당좌표의 숫자를 공개하고 공개된숫자가 15일 경우 게임을 종료한다.

- 해당좌표의 공개여부를 저장하기위해 셔플된 2차원배열[5][5]을 2개 가지는 3차원 배열을 만들어야 한다

```
Scanner scan = new Scanner(System.in);  
|  
//[1][][] -> 공개여부 저장 [0][][] -> 데이터  
// arr[1][0][0] = 1 -> 0,0 좌표는 공개  
int[][][] arr = new int[2][5][5];  
  
//1~25 순차 입력  
for(int i=0;i<arr[0].length;i++)  
    for(int j =0;j<arr[0][0].length;j++)  
        arr[0][i][j] = i*arr[0][0].length+(j+1);
```

```
좌표를 입력하여 숫자 15를 찾으시오.:  
00  
-----  
10      *      *      *      *  
*      *      *      *      *  
*      *      *      *      *  
*      *      *      *      *  
*      *      *      *      *  
-----  
좌표를 입력하여 숫자 15를 찾으시오.:  
12  
-----  
10      *      *      *      *  
*      *      *      *      *  
*      1      *      *      *  
*      *      *      *      *  
*      *      *      *      *  
-----  
좌표를 입력하여 숫자 15를 찾으시오.:
```

3-3 문제풀이 (expert)



THANK YOU



강사 박주병