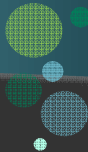


제6장 배열(Array)



7. 2차원 배열

- []의 개수가 차원의 수를 의미한다.

선언방법	선언예
타입 [] 변수이름;	int [] [] score;
타입 변수이름 [] [];	int score [] [];
타입 [] 변수이름 [];	int [] score [];

[표5-3] 2차원 배열의 선언

- 초기화 방법

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

좌측 코드는 5행 5열 2차원 배열을 생성함.

25개의 방이 생성되고 메모리에 총 100바이트 생성함.

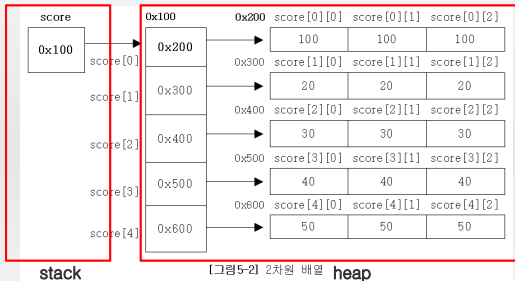
```
int[][] array = new int[][] {  
    {100,50,50},  
    {50,80,50},  
    {100,80,90}  
};
```

좌측 코드는 선언과 동시에 3행 3열의 값을 초기화를 해주고 있다. 메모리에 총 $9 \times 4 = 36$ 바이트가 생성됨.

7. 2차원 배열의 메모리 형태

국어	영어	수학
100[0,0]	100[0,1]	100[0,2]
20[1,0]	20[1,1]	20[1,2]
30[2,0]	30[2,1]	30[2,2]
40[3,0]	40[3,1]	40[3,2]
50[4,0]	50[4,1]	50[4,2]

위의 표는 5행 3열의 표이다. 좌측 부분은 인덱스를 나타내고 있으며, **마지막 인덱스가 [n-1][m-1]의 공식이 성립함**을 알 수가 있다.

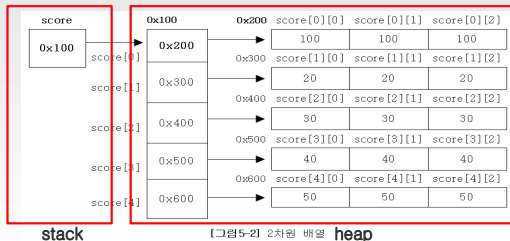


```
for(int i=0; i<score.length; i++) {
    for(int j=0; j<score[i].length; j++) {
        score[i][j] = 20;
    }
}
```

좌측 코드는 위의 메모리 공간에 할당되어 있는 값에 접근하여 모두 20으로 대입하고 있는 코드이다. 여기서 중요한 것은 **2차원 배열에서는 score도 주소 값이며 score[i]값조차도 주소 값이란 것이다.(매우 중요)**

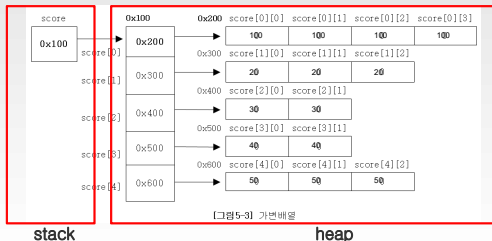
8. 가변 배열(열이 서로 다른 배열)

```
//선언과 동시에 초기화
int[][] score = {
    {100,100,100},
    {20,20,20},
    {30,30,30},
    {40,40,40},
    {50,50,50}
};
```



[그림 5-2] 2차원 배열

```
//선언과 동시에 초기화
int[][] score = {
    {100,100,100,100},
    {20,20,20},
    {30,30},
    {40,40},
    {50,50,50}
};
```



[그림 5-3] 가변 배열

9. 배열의 복사

- for문을 이용한 배열의 복사

```
int[] original = new int[]{10,5,7,8,9};  
int[] cloned = new int[10];  
//for문 복사  
for(int i=0; i<original.length; i++) {  
    //original의 값을 cloned로 대입하고 있다.  
    cloned[i] = original[i];  
}
```

- System.arraycopy()를 이용한 배열의 복사(for문보다 속도가 빠름)

```
//System클래스에 있는 정정메서드 arraycopy이용 복사  
System.arraycopy(original, 0, cloned, 0, original.length);
```

- 배열 복사의 또 다른 방법

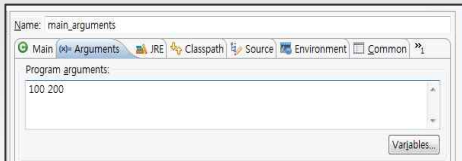
Arrays클래스의 copyOf(), copyOfRange() 이용

10. 사용자 입력 받기(main()메서드)

- main()메서드의 매개변수 값 활용하기

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

좌측 코드는 main(String[] args)이다.
역시, 메서드이기 때문에 이용을 할 수가 있다.
매개변수를 주는 것을 이클립스를 이용하여 사용할 수가 있다.



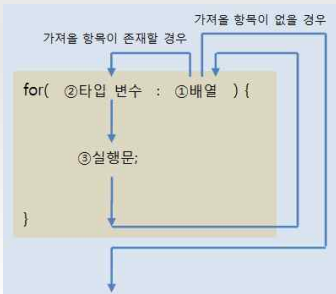
main(String[] args)의 매개변수 값을 설정하기 위하여 메뉴 탭에서 Run->Run Configuration을 선택하여, Arguments 탭에서 값을 지정하면 프로그램이 시작하면서, main(String[] args)에 전달된다. 단, String[] 타입으로 넘어감에 주목하자.

```
String strNum1 = args[0]; //첫번째 데이터 얻기  
String strNum2 = args[1]; //두번째 데이터 얻기
```

위와 같이 값을 전달했다면, String[] 타입으로 넘어오므로, 좌측과 같이 값을 얻어낼 수가 있다.

11. 향상된 for문(JDK 1.5)

- 배열 및 컬렉션의 항목 요소들을 순차적으로 처리
- **인덱스를 사용하지 않고, 바로 항목 요소 반복**



```
int[] scores = { 95, 71, 84, 93, 87 };
```

```
int sum = 0;  
for (int score : scores) {  
    sum = sum + score;  
}
```

12. JAVA API(DOCS) 이용법

* JAVA API 이용하여 메서드 용도를 알자.

<https://docs.oracle.com/javase/8/docs/api/>

자바에서 제공하는 모든 클래스와 메서드들이 담겨 있으므로
즐거찾기에 넣어두고, 자주 활용하자.

감사합니다.

