

# 5강 배열

프로그래밍 강사 : 이태성



# 목차



- 일차원 배열
- 이차원 및 다차원 배열
- 가변 배열
- 배열을 인수로 전달
- 배열을 리턴하는 메서드
- 배열의 메서드

# 01. 일차원 배열



# 배열



- 기본 개념

같은 데이터형 + 변수명 + 순차적인 메모리  
나열

- 참조형

new를 통해 생성

- Array로부터 파생된 참조형

- foreach 사용 가능

# 일차원 배열



- 선언 형식

데이터형[] 배열명;

int[] array\_name;

# 일차원 배열



- 생성과 초기화
  - `int[] array;`  
`array = new int[] { 1, 2, 3 };`  
`array = new int[3] {1, 2, 3};`
  - `int[] array = new int[3] {1, 2, 3};`  
`int[] array = new int[] {1, 2, 3};`
  - `int[] array = {1, 2, 3};`

# 일차원 배열



- Array.Length 속성
- foreach 사용 (읽기 전용)

코드 보기

## 02. 이차원 배열





# 이차원 및 다차원 배열



- 행과 열, 면은 콤마(,)로 구분
- 선언 형식

데이터형[,] 배열명;

데이터형[,,,] 배열명;

# 이차원 및 다차원 배열



- 형식 (1)

```
int[,] array1 = new int[2,2];
```

```
int[,,,] array2 = new int[2, 3, 2];
```

```
int[,] array1 = new int[,] {  
                                {1, 2}, {3,
```

```
4}};
```

```
int[,,,] array2 = new int[,,,] {  
                                {{1, 2}, {3, 4}},  
                                {{5, 6}, {7,
```

```
8}} };
```

# 이차원 및 다차원 배열



- 형식 (2)

```
int[,] array1 = new int[2,2] {  
    {1, 2}, {3, 4}};  
  
int[,,,] array2 = new int[2, 3, 2]{  
    {{1, 2}, {3, 4}, {5, 6}},  
    {{7, 8}, {9, 10}, {11,  
12}}  
};
```

# 이차원 및 다차원 배열



- 형식 (3)

```
int[,] array2 = {  
    {1, 2}, {3, 4}, {5, 6}  
};
```

```
int[,,,] array3 = {  
    {{1, 2}, {3, 4}, {5,  
6}},  
    {{7, 8}, {9, 10}, {11,  
12}}  
};
```

# 이차원 및 다차원 배열



- 이차원과 다차원 배열 사용하는 예
  - int형 이차원 배열

[코드 보기](#)

- 다차원형 문자열 배열

[코드 보기](#)

## 03. 가변 배열



# 가변 배열



- 형식

```
데이터형[][] 배열명;  
int[][] array_name;
```

# 가변 배열



- 사용 예(1)

```
int[][] array = new int[3][];
```

```
array[0] = new int[2];
```

```
array[1] = new int[3];
```

```
array[2] = new int[4];
```



# 가변 배열



- 사용 예(2)

```
int[ ][ ] array = new int[3][ ];  
array[0] = new int[2] { 1, 2 };  
array[1] = new int[3] {3, 4, 5 };  
array[2] = new int[4] {6, 7, 8, 9};
```

또는

```
int[ ][ ] array = new int[3][ ];  
array[0] = new int[] { 1, 2};  
array[1] = new int[] {3, 4, 5};  
array[2] = new int[] {6, 7, 8, 9};
```

# 가변 배열



- 사용 예(3)

```
int[][] array = new int[][]  
{  
    new int[] {1, 2, 3},  
    new int[] {4, 5, 6},  
    new int[] {7, 8, 9}  
};
```

# 가변 배열



- 사용 예(4)

```
int[ ][ ] array = {  
    new int[] {1, 2, 3},  
    new int[] {4, 5, 6},  
    new int[] {7, 8, 9}  
}
```

# 가변 배열



- 가변 배열 사용 예

[코드 보기](#)

- 가변 다차원 배열 사용 예

[코드 보기](#)

## 04. 배열을 함수로 전달



# 일차원 배열을 함수로 전달



- `int[] array = {1, 2, 3, 4};`

```
void func(int[] arr)
{

}
```

```
func( array );
```

코드 보기

# 이차원 배열을 함수로 전달



- 이차원 배열을 참조로 전달

```
int[,] array2 = { {1, 2, 3}, { 4, 5, 6}};  
SetArray(array2);
```

```
void SetArray(int[,] arr)  
{  
}
```

- SetArray(new int[,] { {1, 2, 3}, {4, 5, 6}});

## 05. 배열을 리턴하는 함수





# 배열을 리턴하는 함수



- 배열을 리턴한다는 의미?  
배열이 참조하는 객체의 주소값을  
리턴하는 것임

코드 보기

## 06. 배열의 메서드들



# 배열의 메서드



- 배열  
Array 클래스로부터 파생된 객체
- 배열 초기화 메서드  

```
public static void Clear(  
    Array array,  
    int index,  
    int length  
)
```

코드 보기

# 배열 복사



- 배열 복사 메서드  
`public Object Clone()`

[코드 보기](#)