클릭하세요 자바

2부 자바 기본 다루기

– 9장 배열

최문환

9장 배열

1. 1차원 배열
2. 다차원 배열

1. 1차원 배열

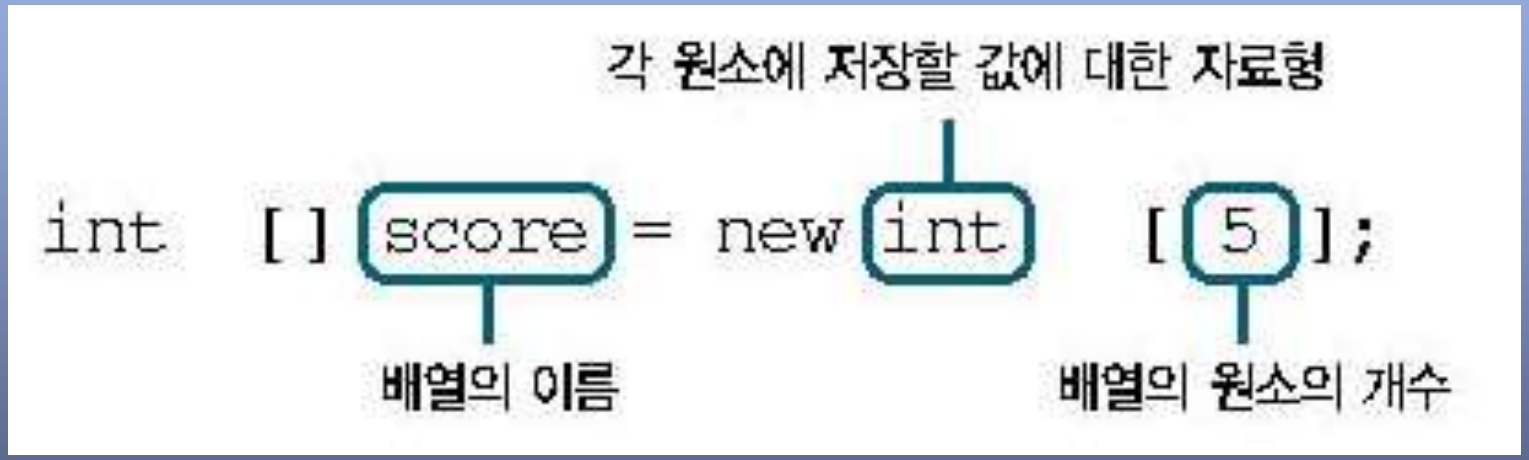
배열은
동일한 자료형의 여러 개의 데이터를 저장하기 위해
여러 개의 방을 한꺼번에 만들어 사용할 수 있는 자료형태

◆ 배열 선언하는 형식은 두 가지 방식

1. new 연산자를 이용하는 방법
2. 해당 배열의 내용을 직접 초기화하는 방법

1.1 new 연산자를 이용하는 방법

<예제> 5명의 점수를 저장하기 위한 정수형 배열



1.1 new 연산자를 이용하는 방법

배열은
메모리(RAM)에 여러 개의 방을 연속된 기억공간으로 할당

score[0]	score[1]	score[2]	score[3]	score[4]
0	0	0	0	0

- 배열은 여러 개의 방들을 동일한 이름(배열명)으로 접근
- 배열에 저장된 데이터는 반복문을 이용하여 일괄처리하여야 데이터처리가 편리
- 배열의 요소(element) :한 명의 점수(정수값)를 저장할 수 있는 기억장소 한 개
- 첨자(색인, index) : 배열의 요소를 각 각 개별적으로 접근하여 데이터를 저장하거나 읽어오기 위해서는 배열의 몇 번째 위치한 기억공간인지를 설명하기 위한 것
- 배열의 첨자는 0번부터 시작하여 번호를 1씩 증가
  - 첫 번째 원소는 score[0], 두 번째 원소는 score[1]로 접근할 수 있습니다. 마지막 원소는 선언된 원소의 개수보다 하나 적은 번호가 되므로 5개의 점수를 저장하기 위한 배열의 마지막 원소를 접근하기 위해서는 score[4]라고 표현

<예제> 1차원의 배열에 값 지정과 출력 방법

```
001:public class G01 {
002:  public static void main(String[] args) {
003:    int []score = new int [5];  //5명의 점수를 저장하기 위한 배열 선언
004:    //배열의 원소에 접근하여 점수를 저장
005:    score[0]=95;
006:    score[1]=70;
007:    score[2]=80;
008:    score[3]=75;
009:    score[4]=100;
010:    //반복문으로 배열을 일괄 처리함
011:    for(int i=0; i<5; i++)
012:      System.out.println( (i+1) + " th score[ " + i + " ] = " + score[i]);
013:  }
014:}
```


1.3 1차원 배열의 다양한 예제

<예제> 총점과 평균 구하기 [파일이름 : Arr01.java]

```
001:public class Arr01 {
002:  public static void main(String[] args) {
003:    int []score = {95, 70, 80, 75, 100};
004:    int total=0;
005:    double ave;
006:    //반복문으로 배열을 일괄 처리함
007:    for(int i=0; i<5; i++)
008:      total += score[ i ];   //총합을 구함
009:    ave = (double) total / 5.0;  //평균을 구함
010:
011:    System.out.println(" Total = " + total);  //총합 출력
012:    System.out.println(" Ave   = " + ave);    //평균 출력
013:  }
014:}
```

1.3 1차원 배열의 다양한 예제

<예제> 5개의 실수값 중에서 최대 값을 구하기 [파일이름 : Arr02.java]

```
001:public class Arr02 {
002:  public static void main(String[] args) {
003:    double []data = {9.5, 7.0, 13.6, 7.5, 10.0};
004:    double max;
005:    //반복문을 수행하기 전에 첫 번째(첨자가 0인) 데이터를 최대값으로
006:    max = data[0];
007:    for (int i = 1; i < 5 ; i++)
008:    if(data[i] > max)   //배열의 원소가 최대값보다 크면
009:      max = data[i];  //새로운 최대값으로 설정
010:    System.out.println("  max = " + max);  //최대값 출력
011:  }
012:}
```


2. 다차원 배열

학생 5명의 국어, 영어, 수학 점수를 처리하기 위해서는 데이터를 다음과 같이 저장할 수 있다.

	국어	영어	수학
1번 학생	85	60	70
2번 학생	90	95	80
3번 학생	75	80	100
4번 학생	80	70	95
5번 학생	100	65	80

2차원 배열은
행렬형태로 데이터를 처리할 때 사용
-행 단위에는 각 학생의 정보가 저장
-열 단위로는 각 과목별 점수가 저장

2. 다차원 배열

3과목을 치룬 학생 5명의 데이터를 저장하기 위해서는 2차원 배열 선언

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

	0열	1열	2열
0 행	score[0][0]	score[0][1]	score[0][2]
1 행	score[1][0]	score[1][1]	score[1][2]
2 행	score[2][0]	score[2][1]	score[2][2]
3 행	score[3][0]	score[3][1]	score[3][2]
4 행	score[4][0]	score[4][1]	score[4][2]

2차원 배열 score는 15(5 x 3)개의 원소를 저장하는 메모리가 할당

3번 학생의 영어 점수를 80점으로 저장하기

```
score[2][1] = 80;
```

2. 다차원 배열

<예제> 2차원 배열에서 요소의 값 출력하기-[파일이름 : Arr03.java]

```
001:public class Arr03 {
002:  public static void main(String[] args) {
003:    //정수값을 담을 수 있는 5행 3열짜리 기억 공간이 생성
004:    int  [][]score=new int [5][3];
005:    int  row, col;  //반복문에서 사용할 제어변수 선언
006:    //행과 열의 위치를 첨자로 지정하여 값 대입
007:    score[0][0]=10;  score[0][1]=90;  score[0][2]=70;
008:    score[1][0]=60;  score[1][1]=80;  score[1][2]=65;
009:    score[2][0]=55;  score[2][1]=60;  score[2][2]=85;
010:    score[3][0]=90;  score[3][1]=75;  score[3][2]=95;
011:    score[4][0]=60;  score[4][1]=30;  score[4][2]=80;
012:
013:    ///반복문으로 일괄처리
014:    for(row = 0; row < 5 ; row++){
015:      for(col = 0; col < 3 ; col++)
016:         System.out.print(" " +score[row][col]);
017:      System.out.println("");  //행단위로 줄 바꿈
018:    }
019:  }
020:}
```