제4장 배열

배열

1장에서 숫자나 문자열을 담을 수 있는 변수를 배웠습니다.

프로그램이 커질 수록 많은 변수를 사용해야 합니다.

많은 데이터의 평균을 구한다던지.

이 장에서 배울 배열은, 변수를 보다 편리하게 사용하기 위한 방법입니다.

배열을 사용함으로써 한번에 많은 변수를 처리할 수 가 있습니다.

리스트 4-1 점수관리 프로그램

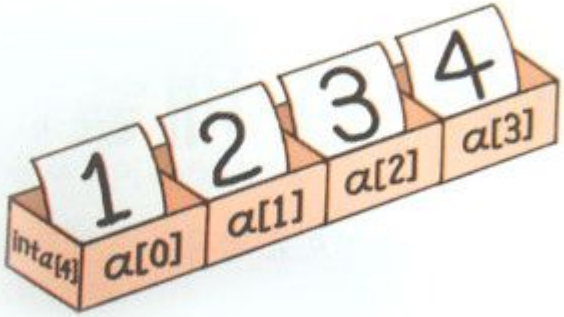

```
int math = 20;
int korean = 30;
int science = 40;
int english = 50;
int society = 80;

int sum = math + korean + science + english + society;

int avg = sum / 5;
System.out.println("합계: " + sum);
System.out.println("평균: " + avg);
```

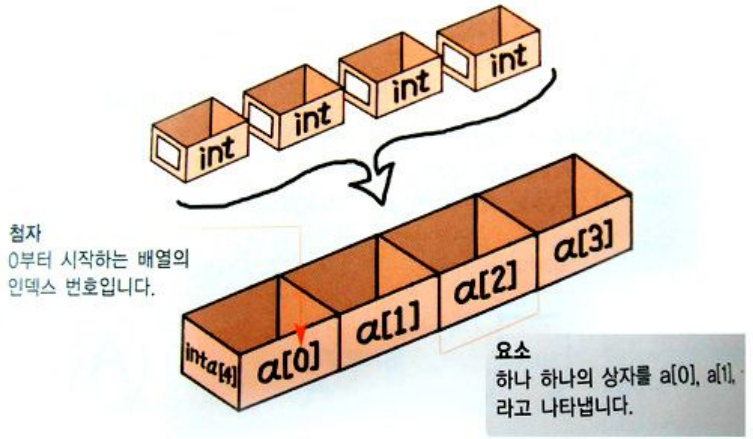
자료구조 (data structure) : 비슷한 데이터를 모아서 담을 수 있는 구조

배열 (array)

배열(array) 이란 동일 종류의 복수 데이터를 순서대로 저장하는 데이터 구조

배열의 최초의 요소는 0번이다





리스트 4-2 배열의 작성 순서

```
1  public class Main {
2      public static void main(String[] args) {
3          int[] score;           // 배열변수의 선언
4          score = new int[5]; // 요소의 작성과 대입
5      }
6  }
```

리스트 4-3 배열의 작성 순서 2

```
1  public class Main {
2      public static void main(String[] args) {
3          int[] score = new int[5];
4      }
5  }
```

리스트 4-4 배열의 길이 구하기


```java
public class Main {
    public static void main(String[] args) {
        int[] score = new int[5];
        int count = score.length;
        System.out.println("요소의 수: " + count);
    }
}
```

리스트 4-5 배열의 요소에 값을 대입


```java
public class Main {
    public static void main(String[] args) {
        int[] score;
        score = new int[5];
        score[1] = 30;          // 두번째 요소에 30 대입
        System.out.println(score[1]);
    }
}
```

리스트 4-6 초기화되어 있지 않은 변수를 이용


```
1  public class Main {
2      public static void main(String[] args) {
3          // 변수를 사용하기 전에는 반드시 초기화를 해야 한다
4          int x;
5          System.out.println(x);  // 컴파일 에러
6      }
7  }
```

리스트 4-7 배열의 초기화


```java
public class Main {
    public static void main(String[] args) {
        // 배열의 요소는 자동으로 초기화된다
        // 5개의 요소가 전부 0으로 초기화
        int[] score = new int[5];
        System.out.println(score[0]);    // 에러가 아님
    }
}
```

생략기법


```
int[] score1 = new int[] { 20, 30, 40, 50, 80 };
int[] score2 = { 20, 30, 40, 50, 80 };
```

배열과 예외

범위를 벗어난 요소를 이용할 때 **예외**(exception) 발생

예외 : 에러

리스트 4-8 점수관리 프로그램 (배열 버전)


```java
public class Main {
    public static void main(String[] args) {
        int[] score = { 20, 30, 40, 50, 80 };
        int sum = score[1] + score[2]
                + score[3] + score[4] + score[5];
        int avg = sum / score.length;
        System.out.println("총점 " + sum);
        System.out.println("평균 " + avg);
    }
}
```

ArrayIndexOutOfBoundsException 발생 : 존재하지 않은 요소를 사용하려고 했다라고 에러의 원인을 판단 가능

배열 첨자 범위외 예외

배열 데이터를 모아서 취급

처음에 비해 코드가 많이 줄어졌다

좀 더 개선 해 보자

과목이 늘어났을 때에 합계를 구할 때 수정 할 부분들이 있다

for문과 조합하여 해결 할 수 있다

리스트 4-9 배열과 for문


```java
public class Main {
    public static void main(String[] args) {
        int[] score = { 20, 30, 40, 50, 80 };
        for (int i = 0; i < score.length; i++) {
            System.out.println(score[i]);
        }
    }
}
```

for each 문

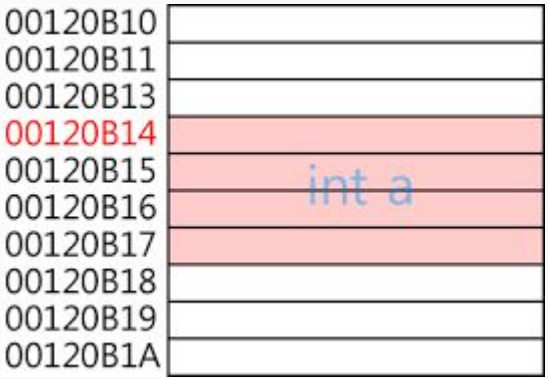

```java
int[] score = { 20, 30, 40, 50, 80 };

// 일반 for 문
for (int i = 0; i < score.length; i++) {
    System.out.println(score[i]);
}
// 향상된 for 문
for (int value : score) {
    System.out.println(value);
}
```

리스트 4-12 실행결과는?


```java
public class Main {
    public static void main(String[] args) {
        int[] a = { 1, 2, 3 };
        int[] b;
        b = a;
        b[0] = 100;
        System.out.println(a[0]);
    }
}
```

메모리와 변수

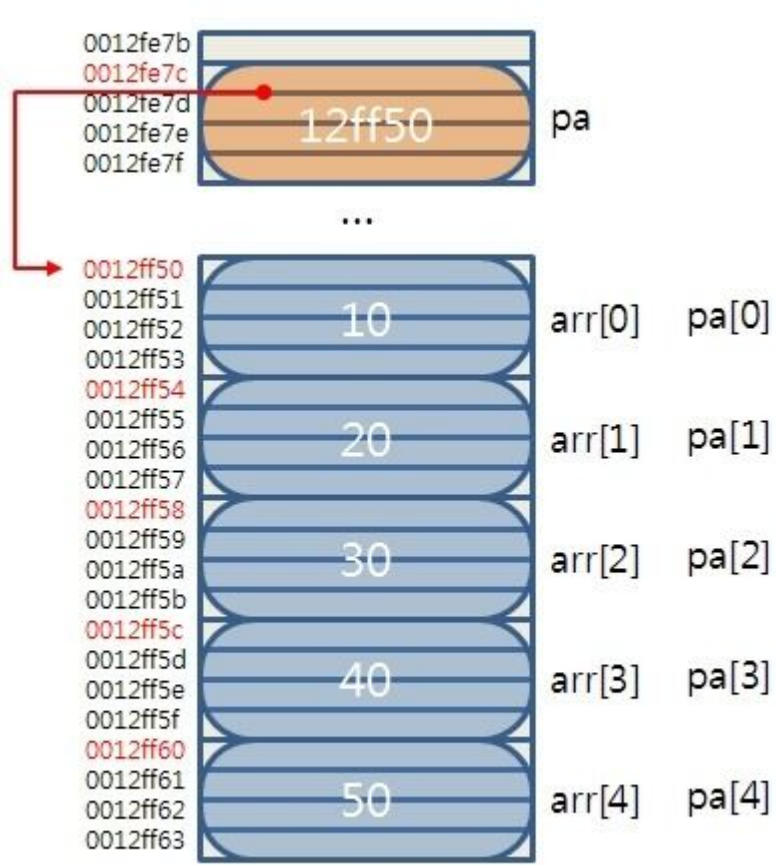
컴퓨터 내부에 변수가 할당된 모습



배열 변수에는 5개의 요소가 들어있는 것이 아니다.
최초의 요소의 주소 (address 또는 reference) 가 대입된다

int[] pa = new int[5]; 를 실행했을 때의 메모리상의 모습

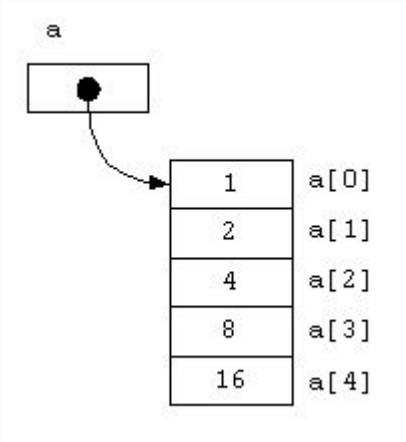
1. int형 요소를 5개 가지는 배열이 메모리상에 작성 됨
2. int[] 형의 배열변수 score 가 메모리 상에 작성됨
3. score 에 배열의 선두요소의 주소가 대입된다



메모리 표현

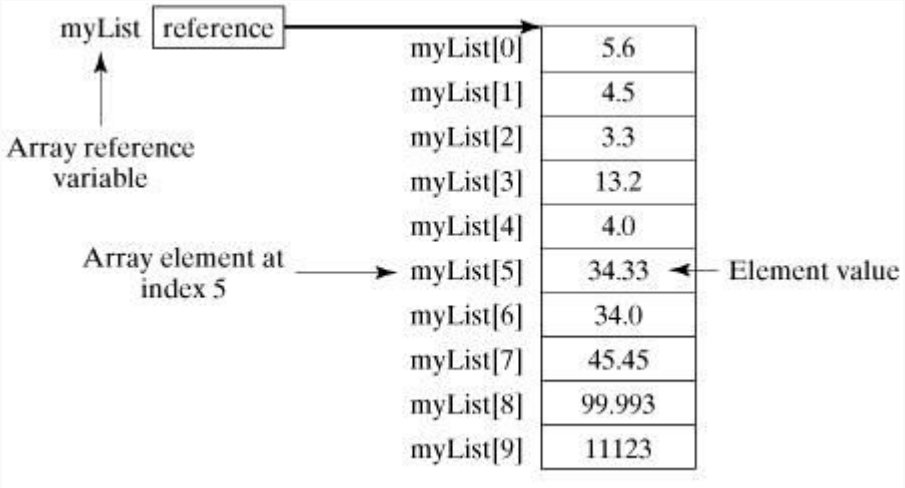
프로그램에서 score[n] 을 지정했을 때

- score 안에 들어있는 주소를 꺼내서, 배열 (선두요소) 를 찾는다
- 찾은 배열의 선두요소로부터 n 개 뒤쪽의 요소를 읽는다.



참조 (reference)

배열처럼 변수명을 지정 했을 때, 그 값이 아니라 주소를 가리키는 것을 **참조** (reference) 라고 한다.
그리고 그 변수를 **참조형** (reference type) 변수라고 한다.
int 나 boolean 같은 “기본형"(primitive type) 변수와 구별 된다.



결론 : 변수 b는 변수 a 와 같은 배열을 참조한다


```java
public class Main {
    public static void main(String[] args) {
        int[] a = { 1, 2, 3 };
        int[] b;
        b = a;
        b[0] = 100;
        System.out.println(a[0]);
    }
}
```

배열의 정리

“참조형” 변수는 “기본형” 과는 다르다고 했다.

배열의 경우도 마찬가지로 조금 다른 점들을 살펴 보자

가비지 컬렉션 (garbage collection)


```java
public class Main {
    public static void main(String[] args) {
        boolean b = true;
        if (b == true) {
            int[] i = { 1, 2, 3 };
        }
    }
}
```

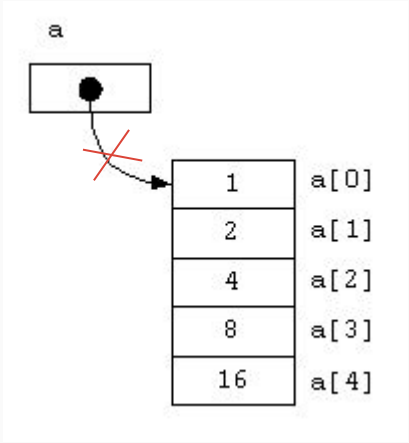
new 로 확보된 요소들은 보통의 변수와 다르기 때문에, 블록이 끝나도 수명이 다하지 않는다

블록 내에서 생성된 배열은 이 후 어떤 방법으로도 읽거나 쓸 수 없고 메모리를 차지하고 있다.
사실상 메모리 내의 쓰레기(garbage) 가 된다.

원래 이렇게 **사용하지 않게 된 메모리는 프로그래머가 정리를 하여야 한다.**
하지만 Java는 **가비지 컬렉션** (GC, garbage collection) 이라는 장치가 더 이상

null

- int[] 같은 참조형 변수에 대입하면, 이 변수는 아무것도 참조하지 않게 됨
- int 형 같은 기본형 변수에 대입할 수 없음



```java
public class Main {
    public static void main(String[] args) {
        int[] a = { 1, 2, 3 };
        a = null;
        a[0] = 10;
    }
}
```

NullPointerException

배열의 length 와 문자열의 length()


```java
public class Main {
    public static void main(String[] args) {
        String s = "Java로 개발";
        System.out.println(s.length());
    }
}
```

String의 length() 는 한글, 영문, 공백 관계없이 1문자로 카운트
배열의 length 와 비슷하지만 () 를 붙여야 한다

다차원 배열

arr[0][0] 1	arr[0][1] 2	arr[0][2] 3
arr[1][0] 4	arr[1][1] 5	arr[1][2] 6
arr[2][0] 7	arr[2][1] 8	arr[2][2] 9

```java
public class Main {
    public static void main(String[] args) {
        int[][] scores = new int[2][3]; // 2행 3열
        scores[0][0] = 40;
        scores[0][1] = 50;
        scores[0][2] = 60;
        scores[1][0] = 80;
        scores[1][1] = 60;
        scores[1][2] = 70;
        System.out.println(scores[1][1]);
    }
}
```

리스트 4-16 배열의 요소 수 표시


```java
public class Main {
    public static void main(String[] args) {
        int[][] scores = { { 10, 20, 30}, { 30, 40, 50} };
        System.out.println(scores.length);
        System.out.println(scores[0].length);
    }
}
```

연습문제 4-1

다음 조건에 맞는 각 배열을 준비하는 프로그램을 작성하시오. 값의 초기화는 필요 없음.

1. int형 값을 4개 담을 수 있는 배열 points
2. double 형 값을 5개 담을 수 있는 배열 weights
3. boolean 형 값을 3개 담을 수 있는 배열 answers
4. String형 값을 3개 담을 수 있는 배열 names

연습문제 4-2

다음 조건에 맞는 프로그램을 작성하시오.

1. 3개의 계좌 잔액 “121902”, “8302”, “55100” 이 담겨 있는 int 형 배열 moneyList 를 선언하시오
2. 그 배열의 요소를 1개씩 for 문으로 꺼내서 화면에 표시하시오
3. 같은 배열 요소를 foreach 문으로 1개씩 꺼내서 화면에 표시하시오

연습문제 4-3


```java
// 각 라인에서 발생하는 예외의 이름을 답하시오
public class Main {
    public static void main(String[] args) {
        int[] counts = null;
        float[] heights = { 171.3F, 175.0F };
        System.out.println(counts[1]);  // 예외 발생
        System.out.println(heights[2]); // 예외 발생
    }
}
```

연습문제 4-4

다음 4개의 조건에 맞는 “숫자 맞추기 퀴즈" 프로그램을 작성 하시오.

1. 3개짜리 int형 배열 numbers를 준비하시오. 이 때 초기화는 각각 3, 4, 9 로 합니다.
2. 화면에 “1자리의 숫자를 입력 해 주세요" 라고 표시합니다
3. `int input = new java.util.Scanner(System.in).nextInt();` 대입합니다

4. input값이 3, 4, 9 중 하나와 같다면 “정답!” 이라고 표시합니다.