고건

Lecture 7 – 배열

배열이란 여러 데이터를 하나로 묶은 것이다. 일반 변수가 하나의 단일 공간이라면, 배열은 일련의 공간을 여러 칸으로 나눈 것이다. 배열은 수많은 데이터를 하나의 변수로 관리함으로써 반복문과 함께 코드의 효율을 높일 수 있다.

배열 안에 담긴 데이터는 타입 유형이 모두 같다. 정수형 배열에는 정수 값만, 실수형 배열에는 실수 값만 담겨야 한다. 한번 지정한 배열의 길이는 바뀌지 않는다. 배열의 길이는 length 키워드를 통해 가져올 수 있다. 배열의 이름이 arr이면 배열의 길이는 arr.length 이다.

배열을 구현하는 방법에는 두 가지가 있다. 먼저 초기화를 통한 생성 방법은 배열 안의 데이터의 초기값을 중괄호 안에 명시하여 만드는 방법이다. 예를 들어 int[] scores = { 88, 98, 68, 72, 47 } 처럼 배열을 구현할 수 있다. 두 번째로는 배열의 길이를 지정하는 방법이다. 예를 들어 double[] grades = new double[4] 이렇게 하면 4칸 길이의 double 타입 배열을 만들어진다. 이때 중요한 것은 4칸 길이의 배열 각각에 들어있는 초기값은 모두 0.0이라는 것이다. 길이를 지정하여 만든 배열은 타입에 따라 초기값이 달라지는데, 정수형 배열은 0, 논리형 배열은 false, 문자열 배열의 초기값은 null 이다. 배열을 쓰기 위해선 배열의 각각의 칸에 하나씩 대입할 수도 있고, 반복문을 통해 순서대로 대입할 수도 있다. 이때 주의해야 할 점은 인덱스의 범위이다. 배열의 인덱스 범위를 벗어난 접근은 에러를 발생시켜, 프로그램이 비정상적으로 종료할 수 있다. 예를 들어 arr[-1]에서 인덱스가 음수이기 때문에 에러가 발생한다.

문자열은 toCharArray() 메소드를 통해 문자의 배열로 바뀔 수 있다. 예를 들어 String str = “ABCD”라면 char[] charArr = str.CharArray();를 통해 charArr = [A, B, C, D]의 배열을 만들 수 있다.

이차원 배열은 기본 배열인 일차원 배열을 묶어 행과 열을 가지는 새로운 배열을 이루는 형태이다. 기본적인 생성 방법은 int[][] scores = {{~},{~}} 이러한 형태이다. 서로 다른 일차원 배열을 묶어 새로운 이차원 배열을 생성할 수도 있다.

동적 배열은 처리할 데이터의 개수가 고정된 경우가 아닐 때 사용한다. ArrayList 객체를 생성하여 만들 수 있다. ArrayList<참조 타입> 참조변수 = new ArrayList<>(); 인데 참조타입에는 기본적으로 integer, Long, Double, Character, Boolean 등을 사용한다. ArrayList 원소에 접근하기 위해서는 참조변수.add(데이터), 참조변수.remove(인덱스번호), 참조변수.get(인덱스번호), 참조변수.size() 이 있다.

문자열 비교하는 연산자 중 ==와 != 연산자는 두 문자열의 내용을 비교하는 것이 아니라 동일한 객체인지 검사하는 연산자이다. 예를 들어 서로 다른 변수 a, b에 같은 값인 1을 할당했다면, a == b의 논리값은 true가 아닌 false가 나온다. 왜냐하면 두 변수의 주소 값이 다르기 때문이다. 문자열을 비교하기 위한 메소드에는 int compareTo(String s), Boolean equals(String s) 등이 있다.

정유진

- Lecture 7 - “정유진의 정리본 ...! 태강아 항상 파이팅 ....!!!!!!!!”

**[배열]**

배열(Array)이란 여러 데이터를 하나로 묶은 것을 말합니다. 일반 변수는 하나의 단일 공간을 차지하고 배열은 이러한 일련의 공간을 여러 칸으로 나눈 것입니다. int[] scores = { 88, 98, 68, 72, 47 }; 이 배열은 5개의 정수를 레퍼러스 변수 scores로 관리합니다. scores가 기본변수와 다른 점은 정수를 직접 가지지 않고, 정수가 저장된 배열을 가리킨다는 것입니다. 이것을 우리는 레퍼런스 변수라 부릅니다.

배열을 사용하면, 수많은 데이터를 단 하나의 변수로 관리할 수 있을 뿐만 아니라 반복문과 함께 코드의 효율을 높일 수 있습니다.

배열에 담긴 데이터는 서로 다른 타입이 아닌 모두 같은 타입 유형을 가집니다. 정수형 배열에는 정숫값, 실수형 배열에는 실숫값만 담겨야 합니다. 또한 배열 속 데이터는 인덱스로 구분합니다. 인덱스(index)란 배열 공간을 번호로 구분한 것으로, 0부터 시작한다는 특징이 있습니다. String[] names = { “Kim”, “Lee”, “Park” };에서 3개 문자열, “Kim”은 0번. “Lee”는 1번, “Park”은 2번에 위치합니다. 마지막으로 배열은 한번 만들어지면 그 길이가 변하지 않습니다. 예를 들면, 배열을 4칸 길이로 만들었다면 최대 4개의 데이터까지만 저장할 수 있습니다. 그리고 이러한 배열의 길이는 length 키워드를 통해 가져올 수 있습니다. 배열의 길이가 N일 때, 인덱스의 범위는 0부터 N-1까지입니다.

배열을 생성할 수 있는 방법에는 2가지가 있는데 첫 번째, 초기화를 통한 생성입니다. 초기화를 통한 생성은 중괄호 안에 초깃값을 명시하여 만드는 방법입니다. int[] scores = { 88, 98, 68, 72, 47 }; 이 배열은 5개의 정숫값으로 배열을 만들고, 이를 레퍼런스 변수 scores에 대입한 코드입니다. 배열의 값들이 정수형이므로 레퍼런스 변수 scores의 타입은 int[]가 되는데, [] 기호가 배열임을 표시해주는 기호입니다. 두 번째 방법은 길이 지정을 통한 생성입니다. new 키워드와 배열의 타입, 길이 정보를 명시해주면 됩니다. double[] grades = new double[4]; 이 배열은 4칸 길이의 double 타입 배열을 만들고, 이를 레퍼런스 변수 grades로 가리키게 하였습니다. 여기에서 가리킬 배열이 실수형이므로 타입은 double[]이어야 합니다. 배열의 초깃값은 모두 0.0인데 길이를 지정하여 만든 배열은 타입에 따라 초깃값이 결정됩니다.

배열은 레퍼런스 변수에 인덱스를 명시하면 그 값을 가져오거나 변경할 수 있습니다. 배열 사용 시 인덱스의 범위를 주의해야 합니다. 배열의 인덱스 범위를 벗어난 접근은 에러를 발생시키고 이는 프로그램의 비정상 종료로 이어집니다.

배열을 활용하면 반복문을 효과적으로 사용할 수 있습니다. 또한 배열은 메소드의 입력변수로 전달될 수도 있습니다.

이차원 배열은 기본 배열(일차원 배열)을 묶어 새로운 배열을 이루는 형태입니다. 즉 이차원 배열은 기본 배열(일차원 배열)을 묶어 새로운 배열을 이루는 형태라 할 수 있습니다. 더 나아가 삼차원 배열은 이차원 배열을 묶어 만든 배열입니다.

처리할 데이터의 개수가 고정된 경우가 아닐 때 정적 배열을 사용하면 자원을 낭비하거나 프로그램을 다시 컴파일하는데 Java는 크기가 유동적인 배열을 지원하기 위하여 ArrayList 클래스를 제공합니다. 우리는 이 ArrayList를 동적 배열이라 합니다.

**[문자열]**  
==와 != 연산자는 두 문자열의 내용을 비교하는게 아닌 동일한 객체인지 검사하는 연산자입니다.

테이블이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명