JAVA
자 료 구 조

Array

1. Array

- 배열 : 데이터 타입이 같은 여러 개의 변수를 한데 모아 묶은 것.

  int[] arr = new int[3] // 기본 자료형

  String[] strings = new String[3] // 참조 자료형

배열을 이루는 3가지 요소

1. 배열을 구성하는 자료형
2. 배열의 이름
3. 배열의 길이

- 배열의 선언 방법

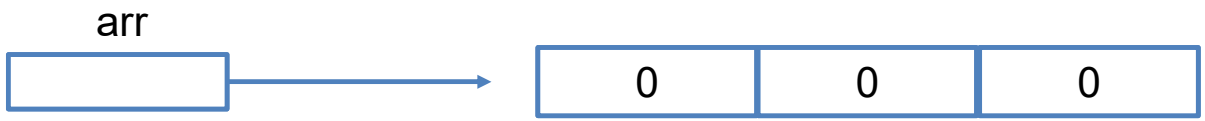
  int[] arr; // 선언방법 1

  int arr[]; // 선언방법 2

JAVA의 배열은 기본적으로 C/C++ 같이 인덱스 0부터 시작 음수 인덱스 사용시 NegativeArraySizeException 발생.

- 배열의 생성과 초기화

  int arr = new int[3];



int 타입 배열의 초기화 모습

1. Array

- 배열의 생성과 초기화

```java
package array;

public class chapter3_inits {
    Run | Debug
    public static void main(String[] args) {
        final int ARRAY_LENGTH = 1;
        String[] strings = new String[ARRAY_LENGTH]; // 참조 타입
        byte[] bytes = new byte[ARRAY_LENGTH];
        short[] shorts = new short[ARRAY_LENGTH];
        int[] ints = new int[ARRAY_LENGTH];
        float[] floats = new float[ARRAY_LENGTH];
        long[] longs = new long[ARRAY_LENGTH];
        double[] doubles = new double[ARRAY_LENGTH];
        boolean[] booleen = new boolean[ARRAY_LENGTH];

        System.out.println("참조 타입의 초기 값: " + strings[0]);
        System.out.println("byte 타입의 초기 값: " + bytes[0]);
        System.out.println("shorts 타입의 초기 값: " + shorts[0]);
        System.out.println("ints 타입의 초기 값: " + ints[0]);
        System.out.println("floats 타입의 초기 값: " + floats[0]);
        System.out.println("longs 타입의 초기 값: " + longs[0]);
        System.out.println("doubles 타입의 초기 값: " + doubles[0]);
        System.out.println("boolean 타입의 초기 값: " + booleen[0]);
    }
}
```

```
ints 타입의 초기 값: 0
floats 타입의 초기 값: 0.0
longs 타입의 초기 값: 0
doubles 타입의 초기 값: 0.0
boolean 타입의 초기 값: false
```

배열 선언 시 초기화 하지 않는 경우
기본 값(default value)로 저장

자료형	기본값
참조 타입	Null
Byte	0
Short	0
Int	0
Float	0.0f
Long	0L
Double	0.0
Char	'\u0000'
Boolean	false

char 형은 출력이 되지 않고 오류 발생.

1. Array

- 배열을 초기화 하는 3가지 방법

```
arr1: [0], length: 1
arr2: [1, 2, 3], length: 3
arr3: [1, 2, 3, 4, 5], length: 5
```

```java
package array;
import java.util.Arrays;

public class chapter3_init {
    Run | Debug
    public static void main(String[] args) {
        int[] arr1 = new int[1]; // (1) 선언과 동시에 0으로 초기화
        int[] arr2 = new int[]{1, 2, 3}; // (2) 배열 생성 시, 크기를 지정하지 않고 저장 할 요소만 명시한다.
        int[] arr3 = {1, 2, 3, 4, 5}; // (3) 저장 할 요소만 명시하는 방법

        System.out.println("arr1: " + Arrays.toString(arr1) + ", length: " + arr1.length);
        System.out.println("arr2: " + Arrays.toString(arr2) + ", length: " + arr2.length);
        System.out.println("arr3: " + Arrays.toString(arr3) + ", length: " + arr3.length);
    }
}
```

1. 선언과 동시에 0으로 초기화 하는 경우
2. 배열 생성 시, 크기를 지정하지 않고 저장할 요소만 명시
3. 저장할 요소만 명시하는 방법

1. Array

- 배열의 반복문

```java
package array;

public class chapter3_loop {
    Run | Debug
    public static void main(String[] args) {
        int[] students = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
        for (int i = 0; i < students.length; i++) {
            System.out.println("학생 번호: " + students[i]);
        }
    }
}
```

```
학생 번호: 1
학생 번호: 2
학생 번호: 3
학생 번호: 4
학생 번호: 5
학생 번호: 6
학생 번호: 7
학생 번호: 8
학생 번호: 9
학생 번호: 10
```

```java
package array;

public class chapter3_loop_foreach {
    Run | Debug
    public static void main(String[] args) {
        int[] students = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
        for (int studentNumber : students) {
            System.out.println("학생 번호: " + studentNumber);
        }
    }
}
```

```
학생 번호: 1
학생 번호: 2
학생 번호: 3
학생 번호: 4
학생 번호: 5
학생 번호: 6
학생 번호: 7
학생 번호: 8
학생 번호: 9
학생 번호: 10
```

두 개의 코드는 같은 코드이나

일반 반복문과 향상된 for문 사용

※ 향상된 for문 : foreach

- 파이썬의 for I in range 기능
- 배열의 연산으로 특정 위치에 값을 삽입,삭제하거나 값을 읽어오는 경우는 사용 불가

초기 및 조건, 증감식을 사용하지 않는 foreach문은
오류 가능성을 줄여주고 반복문 로직에 좀 더 집중 가능

1. Array

```java
package array;
import java.util.Arrays;

public class chapter3_loop_2 {
    Run | Debug
    public static void main(String[] args) {
        int[] students = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
        System.out.println("변경 전: " + Arrays.toString(students));

        int[] students2 = new int[students.length];
        for (int i = 0; i < students.length; i++) {
            students2[(students.length - 1) - i] = students[i];
        }
        System.out.println("변경 후: " + Arrays.toString(students2));
    }
}
```

```
변경 전: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
변경 후: [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

위와 같이 인덱스의 증감을 활용하는 코드는 foreach 사용 불가하나
오른쪽의 최대,최소를 반복문을 통해 출력하는 경우
foreach문을 사용하여 구현이 가능하다.

```java
package array;
import java.util.Random;

public class chapter3_loop_3 {
    Run | Debug
    public static void main(String[] args) {
        Random random = new Random();
        final int ARRAY_LENGTH = 100;

        int[] arr = new int[ARRAY_LENGTH];
        for (int i = 0; i < ARRAY_LENGTH; i++)
            arr[i] = random.nextInt(100);
        }

        int min = 0;
        int max = 0;

        for (int k : arr) {
            if (min > k) {
                min = k;
            } else if (max < k) {
                max = k;
            }
        }

        System.out.println("최소 값: " + min);
        System.out.println("최대 값: " + max);
    }
}
```

```
최소 값: 0
최대 값: 99
```

1. Array

```java
package array;
import java.util.Random;

public class chapter3_loop_4 {
    Run | Debug
    public static void main(String[] args) {
        Random random = new Random();
        final int ARRAY_LENGTH = 10;

        int[] arr = new int[ARRAY_LENGTH];
        for (int i = 0; i < ARRAY_LENGTH; i++) { // 배열의 크기만큼 돈다.
            boolean isPrimeNumber = true;
            int randomValue = random.nextInt(ARRAY_LENGTH); // 난수 생성

            if (randomValue == 1) {
                continue;
            }

            if (randomValue == 2) {
                arr[i] = randomValue;
                continue;
            }

            for (int j = 2; j < randomValue; j++) {
                if (randomValue % j == 0) {
                    isPrimeNumber = false;
                    break;
                }
            }

            if (isPrimeNumber) {
                arr[i] = randomValue;
            }
        }

        System.out.println("[배열에 저장된 소수]");
        for (int k : arr) {
            if (k > 0) {
                System.out.println(k);
            }
        }
    }
}
```

```
[배열에 저장된 소수]
2
2
2
5
5
7
```

N번만큼 반복하며 소수만 배열에 저장하는 프로그램

1과 2의 경우 소수를 확인할 로직 사용할 필요가 없다.

Point) isPrimeNumber 무분별한 반복을 피하기 위한 FLAG

만약 1000 이상의 난수가 생성되었을 때, FLAG를 사용하지 않는다면

소수가 아닌 것이 판단되더라도 j < randomValue 조건을 만족할 때까지 의미없는 반복을 수행한다.

*Appendix 배열의 복사 : 깊은복사와 얕은복사

1. Array

다차원 배열

```
int[][] arr = new int[2][1]
```

행의 크기[가로의 크기]　　열의 크기[세로의 크기]

2차원 배열도 1차원 배열과 같이

1. 선언과 동시에 초기화
2. 생성 후 기본값 초기화가 가능하다.

```
arr의 크기: 3
arr2의 크기: 2
```

```java
package array;

public class chapter3_multiArray {
    Run | Debug
    public static void main(String[] args) {
        int[][] arr = { // 선언과 동시에 초기화
                {1, 2, 3},
                {4, 5, 6},
                {7, 8, 9}
        };

        int[][] arr2 = new int[2][3]; // 생성 후 기본 값 초기화

        arr2[0][0] = 1;
        arr2[0][1] = 2;
        arr2[0][2] = 3;
        arr2[1][0] = 4;
        arr2[1][1] = 5;
        arr2[1][2] = 6;

        System.out.println("arr의 크기: " + arr.length);
        System.out.println("arr2의 크기: " + arr2.length);
    }
}
```

1. Array

	사과	포도	오렌지
월	10000	20000	12000
화	8000	3000	15000
수	20000	15000	38000
목	13000	20000	30000
금	30000	12000	20000
토	35000	30000	25000
일	50000	23000	10000

2차원 배열과 중첩 for문을 이용하여 오른쪽과 같이 구현 가능.

두 번째 for문을 보면 fruitMarket[i]의 길이만큼 반복한다.
만약, fruitMarket의 길이만큼 반복한다면 j=3 시점에
ArrayIndexOutOfboundsException 발생

```
총 합: 439000
사과 평균: 23714
포도 평균: 17571
오렌지 평균: 21428
```

```java
package array;

public class chapter3_multiArray_2 {
    Run | Debug
    public static void main(String[] args) {

        int[][] fruitMarket = new int[][]{
                {10000, 20000, 12000},
                {8000, 3000, 15000},
                {20000, 15000, 38000},
                {13000, 20000, 30000},
                {30000, 12000, 20000},
                {35000, 30000, 25000},
                {50000, 23000, 10000}
        };
        int total = 0;
        int apple = 0;
        int grape = 0;
        int orange = 0;

        for (int i = 0; i < fruitMarket.length; i++) {
            // j의 조건식 주의
            for (int j = 0; j < fruitMarket[i].length; j++) {
                if (j == 0) {
                    apple += fruitMarket[i][j];
                } else if( j == 1) {
                    grape += fruitMarket[i][j];
                } else {
                    orange += fruitMarket[i][j];
                }
                total += fruitMarket[i][j];
            }
        }

        System.out.println("총 합: " + total);
        System.out.println("사과 평균: " + apple / fruitMarket.length);
        System.out.println("포도 평균: " + grape / fruitMarket.length);
        System.out.println("오렌지 평균: " + orange / fruitMarket.length);
    }
}
```

1. Array

2차원 배열을 활용하여 구구단 출력

```java
package array;

public class chapter3_multiArray3 {
    Run | Debug
    public static void main(String[] args) {
        int[][] arr = new int[8][9];

        for (int i = 0, k = 2; i < arr.length; i++, k++) {
            for (int j = 0; j < 9; j++) {
                arr[i][j] = k * (j + 1);
            }
        }

        for (int i = 0; i < arr.length; i++) {
            for (int j = 0; j < 9; j++) {
                if (j != 0 && j % 3 == 0) {
                    System.out.println("");
                }
                System.out.print((i + 2) + "x" + (j + 1) + "=" + arr[i][j])
                System.out.print(" ");
            }
            System.out.println("\n");
        }
    }
}
```

```
2x1=2 2x2=4 2x3=6
2x4=8 2x5=10 2x6=12
2x7=14 2x8=16 2x9=18

3x1=3 3x2=6 3x3=9
3x4=12 3x5=15 3x6=18
3x7=21 3x8=24 3x9=27

4x1=4 4x2=8 4x3=12
4x4=16 4x5=20 4x6=24
4x7=28 4x8=32 4x9=36

5x1=5 5x2=10 5x3=15
5x4=20 5x5=25 5x6=30
5x7=35 5x8=40 5x9=45

6x1=6 6x2=12 6x3=18
6x4=24 6x5=30 6x6=36
6x7=42 6x8=48 6x9=54

7x1=7 7x2=14 7x3=21
7x4=28 7x5=35 7x6=42
7x7=49 7x8=56 7x9=63

8x1=8 8x2=16 8x3=24
8x4=32 8x5=40 8x6=48
8x7=56 8x8=64 8x9=72

9x1=9 9x2=18 9x3=27
9x4=36 9x5=45 9x6=54
9x7=63 9x8=72 9x9=81
```

1. Array

Ex) 2_1

5가지 양수를 입력받은 배열의 모든 요소의 합을 구하는 메서드를 작성하라.

```
public class 2_1 {
    public static void main(String[] args){
        int[] arr = new int[5];
        int sum =0;

        Scanner scanner = new Scanner(System.in);

        // solve
    }
}
```

1. Array

Ex) 2_2

배열의 최댓값과 최솟값을 구하는 메서드를 작성하라.

```
public class 2_2{
    public static void main(String[] args){
        int[] arr = {10,11,2,5,3,3,24,15,6,9};


        // solve
    }
}
```

1. Array

Ex) 2_3

순서가 없는 두 배열에서 서로 같은 요소를 포함하는지 판별하는 메소드를 작성하라.

예) A배열[1,3,2], B배열[2,3,1]은 같은 배열이다.

```
public class 2_3{
    public static void main(String[] args){
        int[] arr1 = {1,3,2};
        int[] arr2 = {2,3,1};


        // solve
    }
}
```

1. Array

Ex) 2_4

중복된 요소를 제거한 새 배열을 반환하는 메서드를 작성하라.

중복 제거된 빈 요소는 기본값 0으로 삽입된다.

```
public class 2_4{
    public static void main(String[] args){

        int[] arr = {5,10,9,27,2,8,10,4,27,1};
        int[] result = new int[10];

        // solve
    }
}
```

1. Array

Ex) 2_5

사용자로부터 소문자 알파벳 1개를 입력받아 대문자로 변경하여 출력하는 코드를 작성하라.

예) ‘a’를 입력받아 ‘A’로 출력한다.

1. Array

Homework

백준 사이트 10818, 2562, 2577, 3052, 1546, 8958, 4344 풀기.

10818. 최소, 최대

2562. 최댓값

2577. 숫자의 개수

3052. 나머지

1546. 평균

8958. OX퀴즈

4344. 평균은 넘겠지