

2021. 12. 27 인지연

1. 배열 기초 + for문

```
public class ArrayTest {  
    public static void main(String[] args) {  
        final int MAX = 100;  
        int[] arr = new int[MAX];  
  
        for (int i = 0; i < MAX; i++) {  
            arr[i] = i + 1;  
            System.out.printf("arr[%d] = %d\n", i, arr[i]);  
        }  
    }  
}
```

배열은 for문과 궁합이 아주 좋다.

배열을 만드는 방법

1. 데이터 타입을 적는다. (int, double..)
2. [] 대괄호를 열고 닫는다.
3. 배열의 변수 이름으로 사용할 이름을 적는다.
4. 메모리 공간을 할당해준다.
 - 4-1. new를 적는다.
 - 4-2. 할당하고자 하는 메모리 데이터 타입을 적어준다.
 - 4-3. 할당할 개수를 [] 대괄호 내부에 적어준다.

[] → index

현재 arr라는 배열은 아래와 같은 형식으로 구성된다.

[1] [2] [3] [] [] [] [] [] [] [] [] [] → arr 배열 공간

0 1 2 3 4 5 6 7 8 9 10 11 12 → 배열순서

여기 0부터 써있는 숫자가 바로 arr[0], arr[1] 을 의미하는 숫자들임

Arr[0] = i+1 = 0+1=1

Arr[1] = 1+1 = 2

Arr[2] = 2+1 = 3

.

.

.

Arr[99] = 99+1 = 100

(0~99라서 100개의 배열이 있는것이다.)

2. Scanner

```
import java.util.Scanner;
```

Scanner 작성 시 import는 꼭 작성해야한다. 프로그램에서 알아서 해줌

```
public class ScannerTest {
```

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

```
        Scanner scan = new Scanner(System.in);
```

스캐너 인스턴스 생성

```
        System.out.print("정수를 입력해보세요: ");
```

```
        // scan.nextInt()를 통해서 정수를 입력 받을 수 있게 도와준다.
```

```
        int num = scan.nextInt();
```

데이터 타입 선택, next(데이터타입) 입력

```
        System.out.println("당신이 입력한 정수는 " + num);
```

```
    }
```

```
}
```

Q. 스캐너 쓸 때 다른 방법은 없나요 ?

A. 이거 말고 안쓴다고 보면 됩니다.

Q. 스캐너 데이터 타입이

float 타입이라면 ? scan.nextFloat()

double 이라면 ? scan.nextDouble()

Scanner : Class 타입의 데이터 타입

(아직 클래스를 모르므로 입력을 수행하기 위해 존재하는 데이터 타입이라고 생각하자.)

new Scanner(System.in)은 시스템으로

입력 들어가는 정보를 해석하겠다 정도로 받아들이면 되겠다.

3. 배열 예문 (1)

피보나치 배열

```
public class FibonacciArray {  
    public static void main(String[] args) {  
        final int MAX = 100;  
        final int START = 2;  
        final int END = 20;  
  
        int[] arr = new int[MAX]; 배열 선언, 배열 값 max로 지정
```

```
        arr[0] = 1;  
        arr[1] = 1; arr[0],[1]은 기존 값이 값을 구할 수 없어서 미리 빼놓음
```

```
        for (int i = START; i < END; i++) {  
            arr[i] = arr[i - 1] + arr[i - 2];  
            System.out.printf("arr[%d] = %d\n", i, arr[i]);  
        }
```

```
    }  
}
```

arr[2] = 2	arr[13] = 377
arr[3] = 3	arr[14] = 610
arr[4] = 5	arr[15] = 987
arr[5] = 8	arr[16] = 1597
arr[6] = 13	arr[17] = 2584
arr[7] = 21	arr[18] = 4181
arr[8] = 34	arr[19] = 6765
arr[9] = 55	
arr[10] = 89	

출력값

For문 집중적으로 보기

Arr[2] = arr[2-1] + arr[2-2] = 1+1 = 2

Arr[3] = arr[3-1] + arr[3-2] = 2+1 = 3

Arr[4] = arr[4-1] + arr[4-2] = 3+2 = 5

Arr[5] = arr[5-1] + arr[5-2] = 5+3 = 8

.

.

Arr[19] = arr[19-1] + arr[19-2] = 4181+2584 = 6,765

4. 배열 예문 (2)

피보나치 + 스캐너 같이 쓰기

```
import java.util.Scanner;

public class AdvancedFibonacciArray {
    public static void main(String[] args) {
        final int START = 2;

        int[] sequence; 배열 이름 선언

        Scanner scan = new Scanner(System.in); 스캐너 인스턴스 생성

        System.out.print("피보나치 수열의 몇 번째 항을 구할까요 ? ");
        int end = scan.nextInt(); 스캐너 입력값 int형으로 적용하기 / 변수이름 end

        sequence = new int[end]; 배열 개수 변수 end를 입력했다. 사용자가 입력하면 그 값이 배열의 개수가 된다.

        sequence[0] = 1; Arr[0][1] 는 for문으로 불가능해서 미리 배열 값 지정
        sequence[1] = 1;

        for (int i = START; i < end; i++) {
            sequence[i] = sequence[i - 1] + sequence[i - 2];
            System.out.printf("arr[%d] = %d\n", i, sequence[i]);
        }
    }
}
```

```
피보나치 수열의 몇 번째 항을 구할까요 ? 10
arr[2] = 2
arr[3] = 3
arr[4] = 5
arr[5] = 8
arr[6] = 13
arr[7] = 21
arr[8] = 34
arr[9] = 55
```

For문 보기

$$S[2] = s[2-1] + s[2-2] = 1+1 = 2$$

$$S[3] = s[3-1] + s[3-2] = 2+1 = 3$$

$$S[4] = s[4-1] + s[4-2] = 3+2 = 5$$

.

.

$$S[9] = s[9-1] + s[9-2] = 34 + 21 = 55$$

```

import java.math.BigInteger;
import java.util.Scanner;

public class MoreAdvancedFibonacciArray {
    public static void main(String[] args) {
        final int START = 2;

        BigInteger[] sequence;

        Scanner scan = new Scanner(System.in);

        System.out.print("피보나치 수열의 몇 번째 항을 구할까요 ? ");
        int end = scan.nextInt();

        sequence = new BigInteger[end];

        // 초항 집어넣을때 아래와 같은 형식으로 넣어줘야 합니다.
        // new BigInteger("집어넣을 숫자") 형식으로 셋팅합니다.
        sequence[0] = new BigInteger("1");
        sequence[1] = new BigInteger("1");

        for (int i = START; i < end; i++) {
            // BigInteger 타입의 변수로 작업해야함
            // 여기서는 sequence이므로 줄여서 seq로 적음
            // seq[i - 1].add(더할 대상)
            // 마찬가지로 더할 대상 또한 BigInteger 타입이어야 합니다.
            sequence[i] = sequence[i - 1].add(sequence[i - 2]);
            System.out.println("arr[" + i + "] = " + sequence[i]);
        }
    }
}

```

5. 배열 예문 (3)

큰 수 출력시 / for문과 배열

21억보다 큰 수를 구하려할때 (21억 이하는 int로 구현 가능)
bigInteger를 쓴다.!

1. 배열 선언 시부터 bigInteger사용
2. Scan.nextInt 할때는 int로 사용한다.
3. 배열 선언 시 new bigInteger (집어넣을 숫자)로 세팅
4. bigInteger의 배열 덧셈 시, s[].add(s[]) 이런식으로 세팅
더할때는 bigInteger끼리만 가능하다

```

arr[47] = 4807526976
arr[48] = 7778742049
arr[49] = 12586269025

```

21억 이상의 값 구할 수 있다.

6. 배열 질문 (1)

** 질문 : arry인덱스를 임의로 하나하나 넣고 싶을땐 어떻게 해야하나요?

```
public class RandomArrayIndex {  
    public static void main(String[] args) {  
        final int ARRMATX = 100;  
        final int MAX = 100;  
        final int MIN = 50;
```

```
        int range = MAX - MIN + 1;  
        int[] arr = new int[ARRMAX];
```

```
        int randIdx, cnt = 1;
```

```
        for (; cnt <= 10; cnt++) {
```

```
            // 임의의 배열 인덱스를 선택하도록 지원
```

```
            randIdx = (int) (Math.random() * ARRMATX);
```

```
            // 실제 임의의 배열 인덱스에 접근하여 랜덤 값을 배치함
```

```
            arr[randIdx] = (int) (Math.random() * range + MIN);
```

```
            System.out.printf("randIdx: %d, arr[randIdx] = %d\n", randIdx, arr[randIdx]);
```

```
        }
```

```
    }
```

```
}
```

질문 :

CNT가 1이라고 FOR문에 지정하면
배열이 순서대로 갔을까요?

왜 비웠는지 잘 모르겠습니다.

랜덤값 생성 준비
배열 선언 / 배열 크기 설정

맨 앞 칸은 비워졌다. CNT가 1이라고 하면 배열이 순서대로 갔을 것이다.
공란으로 뒤서 배열 자체도 랜덤 / 배열 안의 수도 랜덤으로 할 수 있다.

배열의 순서를 랜덤으로 정한다.

랜덤으로 지정한 배열순서안에 랜덤 값을 입력한다.

7. 배열 질문 (2)

질문 : 배열은 왜 쓰나요?

현재 arr라는 배열은 아래와 같은 형식으로 구성된다.

[1] [2] [3] [] [] [] [] [] [] [] [] [] [] → arr 배열 공간

0 1 2 3 4 5 6 7 8 9 10 11 12 → 배열순서

여기 0부터 써있는 숫자가 바로 arr[0], arr[1] 을 의미하는 숫자들임

배열, 값을 하나씩 지정한다.

FOR문을 통해서 순차적으로 배열의 값을 입력해준다.
번거롭게 하나하나씩 입력하지 않아도 된다.