break문 과 continue

- break : 자신이 속한 반복문을 탈출한다.

  만약 반복문이 중첩되어 있을 경우 자신이 속한 하나의 반복문만 탈출한다.

- continue : 특정 조건시 반복을 건너 뛸 때 사용한다.

  continue문을 만나면 반복문 끝으로 이동하기 때문에

  continue 아래의 문장들이 있어도 실행하지 않고 다음 반복으로 넘어간다.

while문

조건식이 **true 일 동안** {}안의 문장을 반복

```
while(조건식) {
    조건식의 연산결과가 true일 동안 반복한 문장
}
```

- 만약 조건식이 처음부터 false일 경우 한번도 실행되지 않을 수도 있다.

do ~ while 문

선실행 후판단

while문과 같지만 **최소한 한번은 실행**된다.(사용자에게 입력을 받을 때 주로 사용)

```
do {
    조건식의 연산결과가 참일 동안 반복할 문장
} while(조건식);
```

- 무한반복

```
while(true) {
    무한반복할 문장
}
```

 # 배열

같은 자료형의 **여러 변수를 하나의 묶음으로 나열** 해 놓은 것

1. 여러개의 저장공간을 배열을 통해 한번에 선언하여 관리한다.
2. 규칙성이 없는 값에 규칙성을 부여한다.(index 번호 자동부여)
3. [ ] 대괄호를 사용한다.
4. 배열은 0부터 시작

- 배열의 선언 (배열을 다루기 위한 **참조변수(배열명)가 선언된 상태**)

```
자료형[ ] 배열명;

int[ ] arData;
```

- 배열의 생성(실제 **저장공간을 생성한 상태**)

```
배열의 이름 = new 자료형[칸수];

arData = new int[5];

자료형[ ] 배열명 = new 자료형[칸수];

int[ ] arData = new int[5];

자료형[ ] 배열명 = {값1, 값2, 값3, ...};

int[ ] arData = {1,2,3,4,5};
```

- 배열의 사용

```
배열명[idx] = 값; // 저장공간

sysout(배열명[idx]); //값
```

- 배열의 index 번호

각 배열의 요소에 자동으로 붙는 번호

인덱스의 범위 0 ~ 배열길이-1

- length (배열의 길이 구하기)

```
배열명.length    // 정수
```

※배열의 인덱스가 0부터 시작하는 이유는 메모리 주소가 0부터 시작하기 때문이다

실제 값들이 저장되어 있는 공간이 만들어 지면 이름은 존재하지 않고 주소값만 존재한다.

따라서 주소값을 참조변수(배열명)가 가지고 있고 주소연산을 통해 다음 주소값으로 이동한다.

시작 주소에서 이동한 횟수가 인덱스 번호(방번호)가 되고, 첫번째 방은 배열명이 가지고 있는 주소에서 0번 옮겨야 하므로 인덱스 번호는 0부터 시작한다.

2차원 배열

배열 안의 배열(가로 행, 세로 행)

1차원배열의 배열(1차원 배열 여러개를 하나로 묶어놓은것)

int [ ][ ] number = new int[2][3]

1) 2차원 배열에서 첫번째 대괄호로 접근한 행들은 주소값이 되고,

두번째 대괄호로 접근하는 열들이 값이 된다.

2) number.length는 행을 나타내고, 열의 길이는 number[행의 idx].length

3) 1차원 배열을 여러개 선언시 관리가 힘들기 때문에 2차원 배열을 한번 선언한다.
하지만 2차원 배열은 메모리낭비가 심하므로 선호하지 않는 편이다.

- 2차원 배열 선언

자료형[ ][ ] 배열명= new 자료형[행][열];

int[ ][ ] number = new int[2][3];

자료형[ ][ ] 배열명= {

{값1, 값2, 값3},

{값4, 값5, 값6}

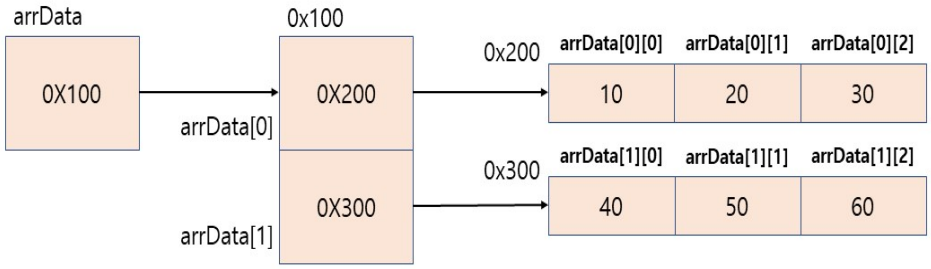
}

각각 **소배열들은 행**을 나타내고 그안에 값들의 **방번호는 열**을 나타낸다.

2차원 배열을 사용할 때에는 두번 접근을 해야하기 때문에 **대괄호가 두개**이다.

2차원배열의 구성

```
int[][] arrData = { { 10, 20, 30 },{ 40, 50, 60 } };
```



2차원 배열

int[][] number= new int[2][3];

행 index
행의 길이 - 1

number[0][0]	number[0][1]	number[0][2]
number[1][0]	number[1][1]	number[1][2]

열 index
(열의 길이 – 1)

- 2차원 배열의 길이

number.length?

number[0].length?