이름 붙는 반복문

1. break문은 가장 근접한 단 하나의 반복문만 벗어날 수 있기 때문에, 여러 개의 반복문이 중첩된 경우에는 break문으로 중첩 반복문을 완전히 벗어날 수 없다.
이때는 **중첩 반복문 앞에 이름을 붙이고 break문과 continue문에 이름을 지정해 줌으로써 하나 이상의 반복문을 벗어날 수 있다.**

2. 사용 형식

```
exit_for: //for문 앞 또는 위에 이름 있는 반복 레이블명을 임의로 지정한다.
for(초기치; 조건식; 증감식){
 for(초기치; 조건식; 증감식) {
   break exit_for;//이중 반복문을 완전히 벗어난다.
 }
}
```

배열의 특징과 생성법

가. 배열이란 **동일한 타입의 여러개의 자료를 한꺼번에 저장하기 위해서 사용하는 것을 배열이라 한다. 배열은 고정된 크기이다.**

나. 배열 생성법

new 키워드를 이용한 배열 생성법

형식) int[] score=new int[3]; //배열 크기가 3인 정수형 배열 score생성

배열 요소 초기화를 통한 배열 생성법

형식) int[] score={10,20,30}; //배열 요소값을 초기화 하면서 배열 크기가 3인 //배열 score생성

다. 배열 길이가 **0인 배열도 생성 가능**하다. **배열의 길이는 int범위의 양의 정수(0도 포함)이어야 한다.**

라. **배열 길이는 배열이름.length를 통해서 알수 있다.**

마. **배열 주소 인덱스 번호값은 0부터 시작한다.**

다차원 배열

1. 2 차원 배열의 선언 방법

```
int[][] score;  int score[][];   int[] score[];
```

위 3가지 방법중 하나를 사용하면 된다.

2. 만약 2차원 배열의 테이블 형태의 4행 3열의 데이터를 담기 위한 배열을 생성하려면 다음과 같이 한다.

형식) int[][] score=new int[4][3];

score[0][0]	score[0][1]	score[0][2]
score[1][0]	score[1][1]	score[1][2]
score[2][0]	score[2][1]	score[2][2]
score[3][0]	score[3][1]	score[3][2]

3. 2차원 배열 초기화 방법

```
int[][] num={
        {10,20},
        {20,30}
   };//2행*2열 2차원 배열구조가 요소값이 초기화 되면서 만들어 진다.
```

자바 클래스 구성 요소

가. 자바 클래스는 **class 키워드로 정의**한다.

나. 클래스 구성 요소

```
class Test{ //자바 한줄 주석문 기호
    int a=10; // 변수 선언부(멤버 변수)

    생성자 영역 부분;
    사용자 정의 메서드;
}
```

다. 사용자 정의 메서드 생성 문법

메서드는 어떠한 **기능 즉 동작을 하게 하는 부분**이다.

형식)

```
public void p(){//void는 반환값이 없는 자료형 즉 리턴 타입이 없다는 뜻.
실행 문장;
}//p()가 사용자 정의 메서드 명
```

//사용자 정의란 자바 개발자에 의해서 임의로 정의 가능한 이름을 뜻한다. 즉 식별자
//할 수 있다.

형식)

```
String getStr(){
  //void형이 아닌 메서드가 정의 된 경우는 return 에 의해서 getStr() 메서드
 //를 호출한 곳으로 결과 문자열을 되돌려 줘야 한다.즉 반환 해야 한다.
  String result=반환 문자열값;
  return result;
}
```

자바 클래스 정의와 객체 생성법

가.**자바 클래스 생성 문법**

형식) **class 클래스 이름{**

```
        변수 선언;

        생성자(){}

        public 자료형(타입) 메서드명(){
      }
  }
```

라고 먼저 **클래스를 정의하고** 난 이후

나.**new 키워드로 새로운 객체명**을 생성한다.

형식) 클래스명 **객체명**=**new** 클래스명();

다.

```
class Test{
      int a;

      void p(){
        실행문장;
      }
  }//Test class end

  Test t=new Test();
  Test t2=new Test();
```

//new 연산 키워드로 생성된 **객체명 t, t2는 서로 다른 객체 주소를 가진다**. 그러므
//로 t.a=7; 로 변경된 멤버 변수 a의 7값은 t2.a로 접근해서 출력하면 7이 아닌
//초기화 된 0으로 출력된다. 이유는 **서로 다른 객체 주소를 가지기 때문에 각각 다른**
//**값을 가지게 되는 것이다.**