■ 재귀함수

재귀호출 (recursive call)
: 매서드의 내부에서 매서드 자신을 다시 호출하는 것을 재귀호출이라고 한다

```
//재귀호출 형식
void method(){
     method();
}
```

■ 재귀함수 작성시 주의 사항 !! 탈출조건명시 (무한loop)

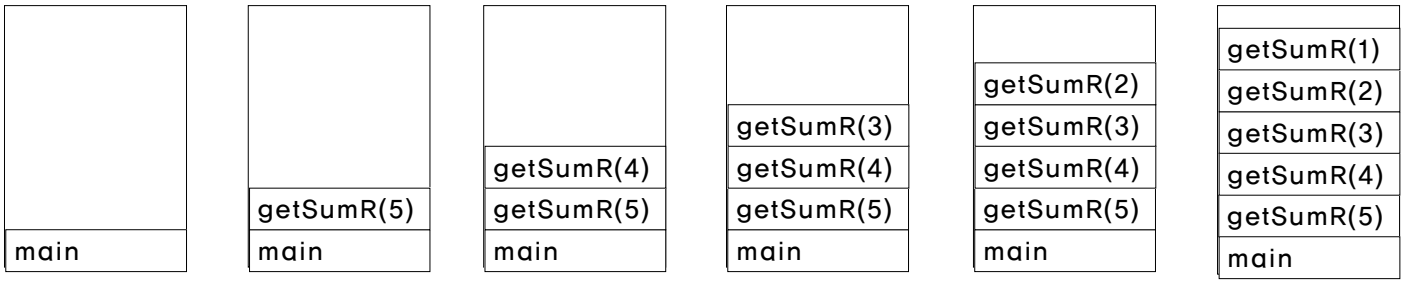
```
// 10까지의 합
void method( int n){
    if( n==1)
      return 1;
    return  method(n-1) +n ;
}
```

//1~5까지의 합

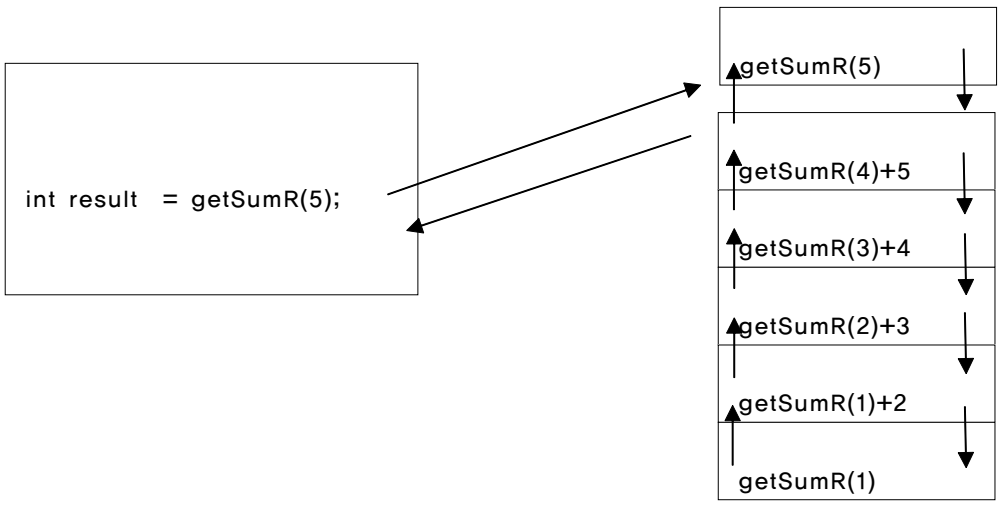
```
int getSum( int n) {
  int sum=0;
  for( int i=1 ; i<=n; i++)
        sum +=i;

  return  sum;
}
```

```
int getSumR( int n) {
    if( n==1)
       return  1;
    return getSumR(n-1)+n;
}
```

```
int getSumR( int n) {
    if( n==1){
       return 1;
    }
    else{
      return getSumR(n-1)+n;
   }
}
```



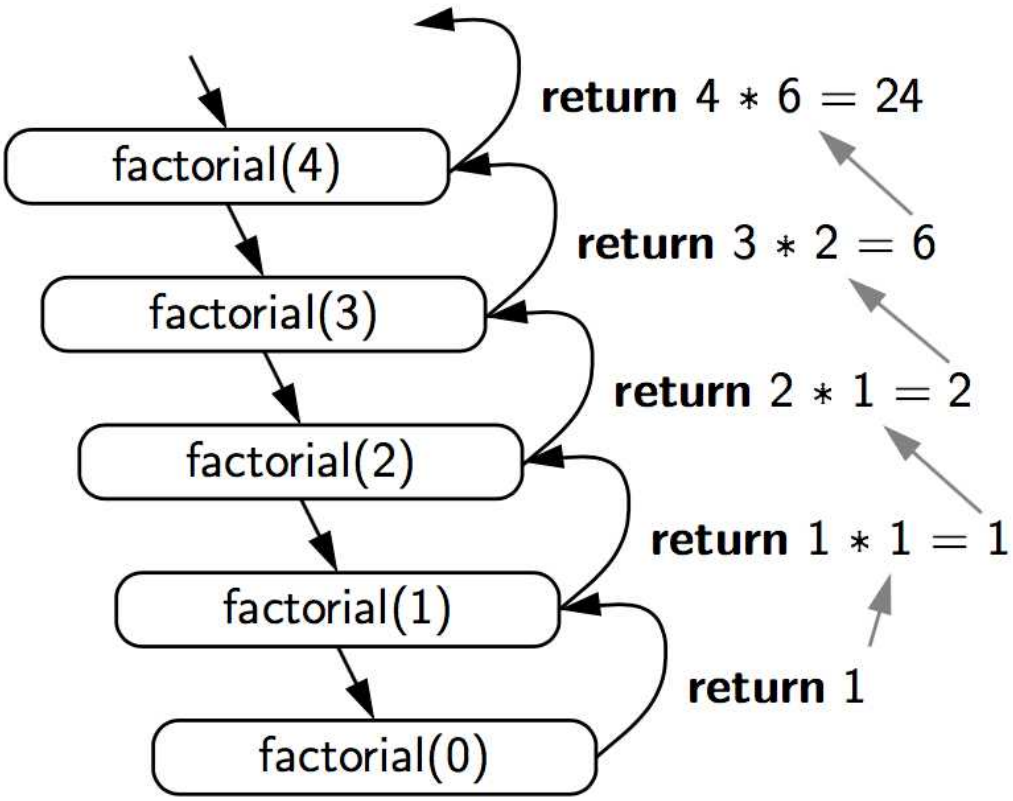
함수호출시 call stack 모습



■ Factorial 구하기

5! =〉1*2*3*4*5 1~5까지의 곱 (Factorial 어렵지 않아요 !)

factorial(5)



■ 배열의 최대값 구하기

```
public class MaxR {
    public static void main(String[] args) {
        int[] array = {3, 7, 1, 9, 4, 6, 8, 2, 5};
        int maxValue = findMax(array, 0, array.length - 1);
        System.out.println("최대값: " + maxValue);
    }

    // 재귀 호출을 사용하여 최대값을 찾는 메소드
    public static int findMax(int[] array, int start, int end) {
        if (start == end) {
            // 배열의 크기가 1이면 해당 원소가 최대값
            return array[start];
        } else {
            // 배열을 두 부분으로 나누어 최대값을 비교하여 반환
            int mid = (start + end) / 2;
            int leftMax =   findMax(array, start, mid);
            int rightMax = findMax(array, mid + 1, end);
            return Math.max(leftMax, rightMax);
        }
    }
}
```