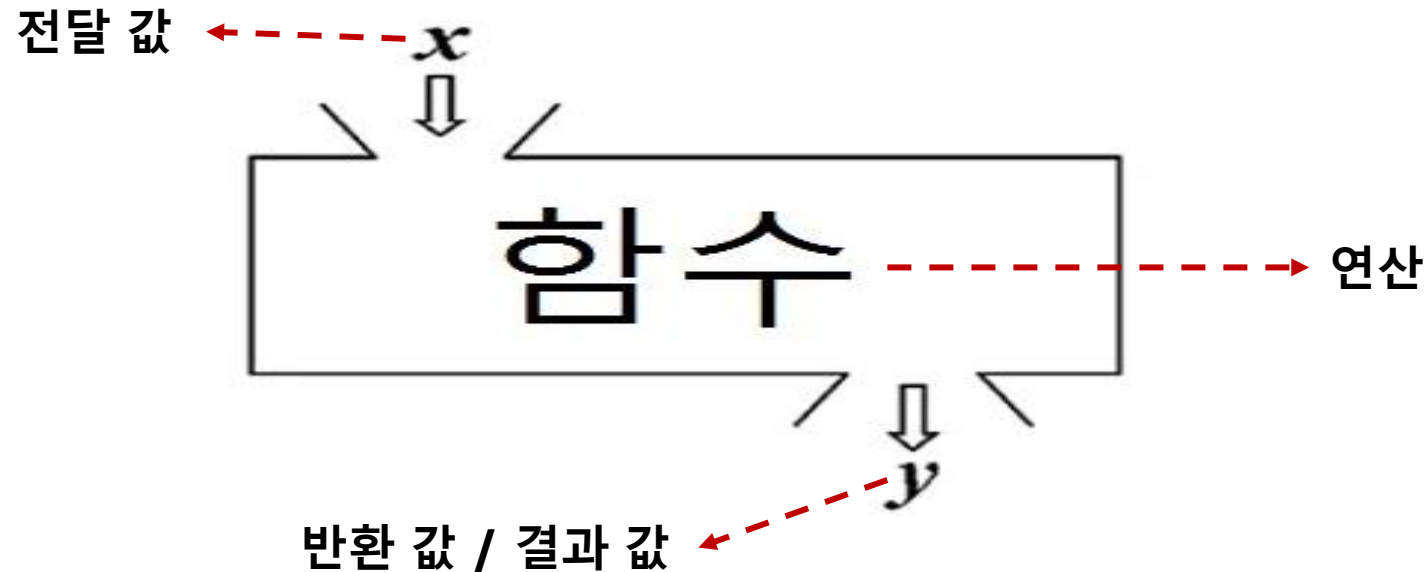


메소드(Method)

▶ 메소드 (Method)

✓ 메소드란 ?

수학의 함수와 비슷하며 호출을 통해 사용, 전달 값이 없는 상태로 호출을 하거나
어떤 값을 전달하여 호출을 하며, 함수 내에 작성된 연산 수행
수행 후 반환 값 / 결과 값은 있거나 없을 수 있음



▶ 메소드 (Method)

✓ 메소드 표현식

```
[접근제한자] [예약어] 반환형 메소드명( [매개변수] ) {
```

```
    // 기능 정의
```

```
}
```

```
public void information() {  
    System.out.println(studentNo);  
}
```

▶ 메소드 (Method)

✓ 메소드 접근제한자

구분		클래스	패키지	자손 클래스	전체
+	public	○	○	○	○
#	protected	○	○	○	
~	(default)	○	○		
-	private	○			

▶ 메소드 (Method)

✓ 메소드 예약어

구분	전체
static	static 영역에 할당하여 객체 생성 없이 사용
final	종단의 의미, 상속 시 오버라이딩 불가능
abstract	미완성된, 상속하여 오버라이딩으로 완성시켜 사용해야 함
synchronized	동기화 처리, 공유 자원에 한 개의 스레드만 접근 가능함
static final (final static)	static과 final의 의미를 둘 다 가짐

▶ 메소드 (Method)

✓ 메소드 반환형

구분	전체
void	반환형이 없음을 의미, 반환 값이 없을 경우 반드시 작성
기본 자료형	연산 수행 후 반환 값이 기본 자료형일 경우 사용
배열	연산 수행후 반환 값이 배열인 경우 배열의 주소값이 반환
클래스	연산 수행후 반환 값이 해당 클래스 타입의 객체일 경우 해당 객체의 주소값이 반환 (클래스 == 타입)

▶ 메소드 (Method)

✓ 메소드 매개변수

구분	전체
()	매개변수가 없는 것을 의미
기본 자료형	기본형 매개변수 사용 시 값을 복사하여 전달 매개변수 값을 변경하여도 본래 값은 변경되지 않음
배열	배열, 클래스 등 참조형을 매개변수로 전달 시에는 데이터의 주소 값을 전달하기 때문에 매개변수를 수정하면 본래의 데이터가 수정됨(얕은 복사)
클래스	
가변인자	매개변수의 개수를 유동적으로 설정하는 방법으로 가변 매개변수 외 다른 매개변수가 있으면 가변 매개변수를 마지막에 설정 * 방법 : (자료형 ... 변수명)

* 매개변수의 수에 제한이 없다.

▶ 메소드 (Method)

- ✓ 메소드 표현식 – 매개변수가 없고 리턴 값이 있을 때

```
[접근제한자] [예약어] 반환형 메소드명() {
```

```
    // 기능 정의
```

```
}
```

```
public int information() {  
    return studentNo;  
}
```


▶ 메소드 (Method)

- ✓ 메소드 표현식 – 매개변수가 없고 리턴 값이 없을 때

```
[접근제한자] [예약어] void 메소드명() {
```

```
    // 기능 정의
```

```
}
```

```
public void information() {  
    System.out.println(studentNo);  
}
```

▶ 메소드 (Method)

- ✓ 메소드 표현식 - 매개변수가 있고 리턴 값이 있을 때

```
[접근제한자] [예약어] 반환형 메소드명(자료형 변수명) {  
    // 기능 정의  
}
```

```
public String information(String studentName) {  
    return studentNo + " " + studentName;  
}
```

▶ 메소드 (Method)

- ✓ 메소드 표현식 - 매개변수가 있고 리턴 값이 없을 때

```
[접근제한자] [예약어] void 메소드명(자료형 변수명) {
```

```
    // 기능 정의
```

```
}
```

```
public void information(String studentName) {  
    System.out.println(studentNo + " " + studentName);  
}
```

▶ getter와 setter 메소드

✓ setter 메소드

필드에 변경할 값을 전달 받아서 필드 값을 변경하는 메소드

✓ 표현식

```
[접근제한자] [예약어] void set필드명(자료형 변수명) {  
    (this.)필드명 = 자료형 변수명;  
}
```

```
public void setStudentNo(int studentNo) {  
    this.studentNo = studentNo;  
}
```

▶ getter와 setter 메소드

✓ getter 메소드

필드에 기록된 값을 읽어서 요청한 쪽으로 읽은 값을 넘기는 메소드

✓ 표현식

```
[접근제한자] [예약어] 반환형 get필드명() {
```

```
    return 필드명;
```

```
}
```

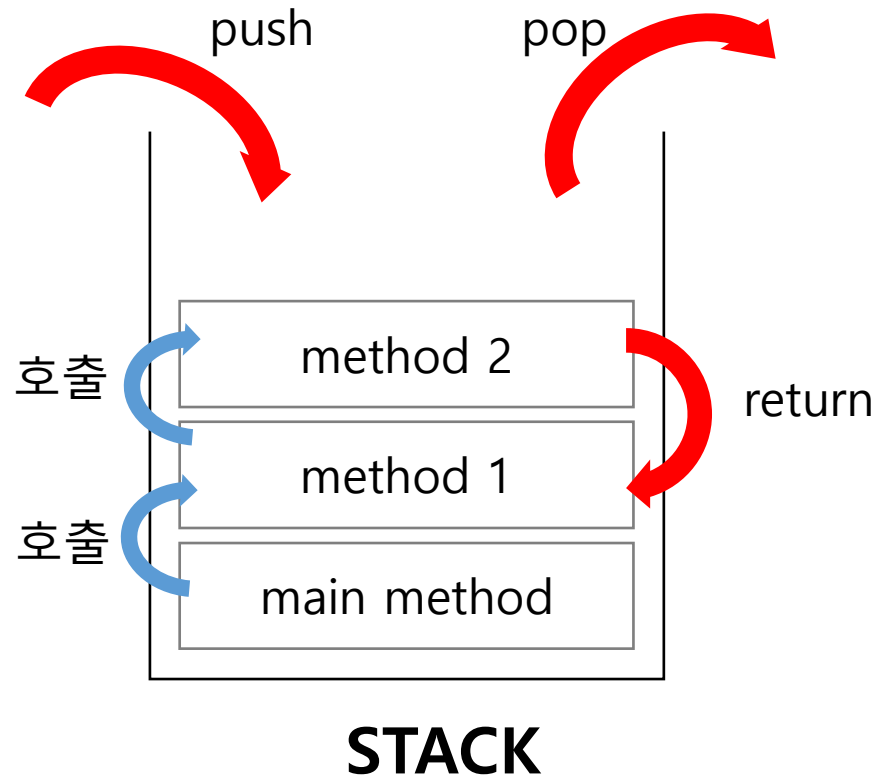
```
public int getStudentNo() {
```

```
    return studentNo;
```

```
}
```

▶ return

해당 메소드를 종료하고 자신을 호출한 메소드로 돌아가는 예약어
반환 값이 있다면 반환 값을 가지고 자신을 호출한 메소드로 돌아감



* STACK의 자료구조 : LIFO(Last-Input-First-Out)