

* 전위 연산자 (+, -, ++, --)

↑
예) $\frac{-a}{+a}$

int a = 100;

++a;

→ a = $\frac{a}{100} + 1$; → bytecode

→ 다음 라인 실행 시
1번 증가

a? ⇒ 101

a = a + 1;

l-value = left-value

↓
메모리

right-value = r-value

↓
값

a = 100 + 1;

a = 101;

int a = 100;

--a;

→ a = $\frac{a}{100} - 1$;

a? ⇒ 99

* 주의!
항상 r-value가 먼저 실행된 후
l-value가 실행된다.

* 후의 연산자 (++, --)

```
int a = 100;
a++;
```

↳ { int temp = a;;
 a = a + 1; }

a? ⇒ 101

```
int a = 100;
a--;
```

↳ int temp = a;;
 a = a - 1;

a? ⇒ 99

```
int a = 100;
a = a++;
```

a? 100!

↳ { int temp = a;;
 a = a + 1;;
 a = temp; }

```
int a = 100;
int b;
```

b = ++a;

← 카운트할 시 ↓ 변한

a = a + 1;

b = a;

a? 101

b? 101

```
int a = 100;
int b;
```

b = a++;

← 카운트할 시 ↓ 변한

int temp = a;

a = a + 1;

b = temp;

a? 101

b? 100


```
int i = 100;
System.out.println(i++);
```

①
②
메서드 호출 전의
파라미터로 주어진
표현식을 먼저 실행한다

```
int i = 2, 3, 4, 5;
int result = (i++) + (i++) * (i++);
```

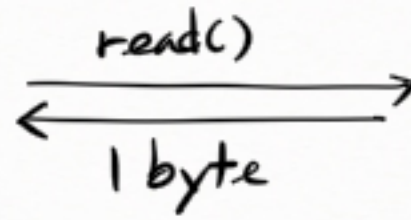
2 + 12 = 14

```
{
  int temp = i;
  i = i + 1;
  System.out.println(temp);
}
```

~~int x = 100;~~
~~(++x)++;~~
~~101~~

* >1바이트 읽기 다지기

14바이트
읽기
다지기
이제 시작.



System.in

← 키보드 입력 들어

- read()
- read(byte[])
- ...

System.out

- print()
- println()
- printf()
- ...

↑ 콘솔 출력 들어

