```
Mirror object to mirror object

Mirror object to mirror object

Deration == "MIRROR X":

Irror mod use x = True

Irror mod use y = False

Operation == "MIRROR Y

Irror mod use x = False

Operation use y = True

Irror mod use y = True

Irror mod use y = True

Irror mod use z = False

Operation == "MIRROR Z"

Operation == "MIRROR Z"
```

rror\_mod.use\_x = False rror\_mod.use\_x = True

#### Java Fundamental

객체 지향 프로그래밍

int("please select exacti

OPERATOR CLASSES ----

ypes.Operator):
 X mirror to the selector
ject.mirror\_mirror\_x"
 ror X"

ontext):
oxt.active\_object is not

### 가변인자(Variable arguments)

- 메서드에 전달하는 **인수의 개수를 가변적**으로 설정
- 메서드 내에서 배열처럼 사용
- 메서드에서 한 번만 정의 가능
- 반드시 마지막 매개변수로 정의

```
[반환형] [메서드명]([자료형]... [매개변수명]) {
// 실행할 내용
}
```

## 메서드 오버로딩(Overloading)

- 메서드명은 같지만 매개변수를 다르게 해서 메서드를 **중복 정의**하는 것
- 같은 이름을 가지나 다르게 동작하도록 만듦
- 반환 값의 자료형은 오버로딩에 영향을 주지 않음
  - ▶두 메서드의 메서드명과 매개변수가 일치하면 같은 메서드로 인식

#### 생성자(Constructor)

- 객체 생성 시 자동으로 호출되는 코드 블록
- 주로 멤버 필드의 초기값을 설정할 때 사용
- 매개변수로 값을 전달받아 필드를 초기화
- 따로 정의하지 않을 경우 기본 생성자가 자동 생성됨

```
[클래스명]([자료형] [매개변수명]) {
// 실행할 내용
}
```

#### this 키워드

- 객체 자신이 갖고 있는 요소를 선택할 때 사용
- 주로 클래스의 요소명과 매개변수명이 같을 때 사용

# 생성자 오버로딩(Overloading)

- 생성자명(=클래스명)은 같지만 매개변수를 다르게 해 서 생성자를 **중복 정의**하는 것
- 다양한 방식의 초기화를 요구할 때 사용

### this() 생성자

- 생성자 내에서 다른 생성자 호출할 때 사용
- this([전달인자1], [전달인자2], ...) 형태로 사용
- 코드의 중복을 피하기 위해 사용
- 반드시 생성자의 첫 줄에서 작성