

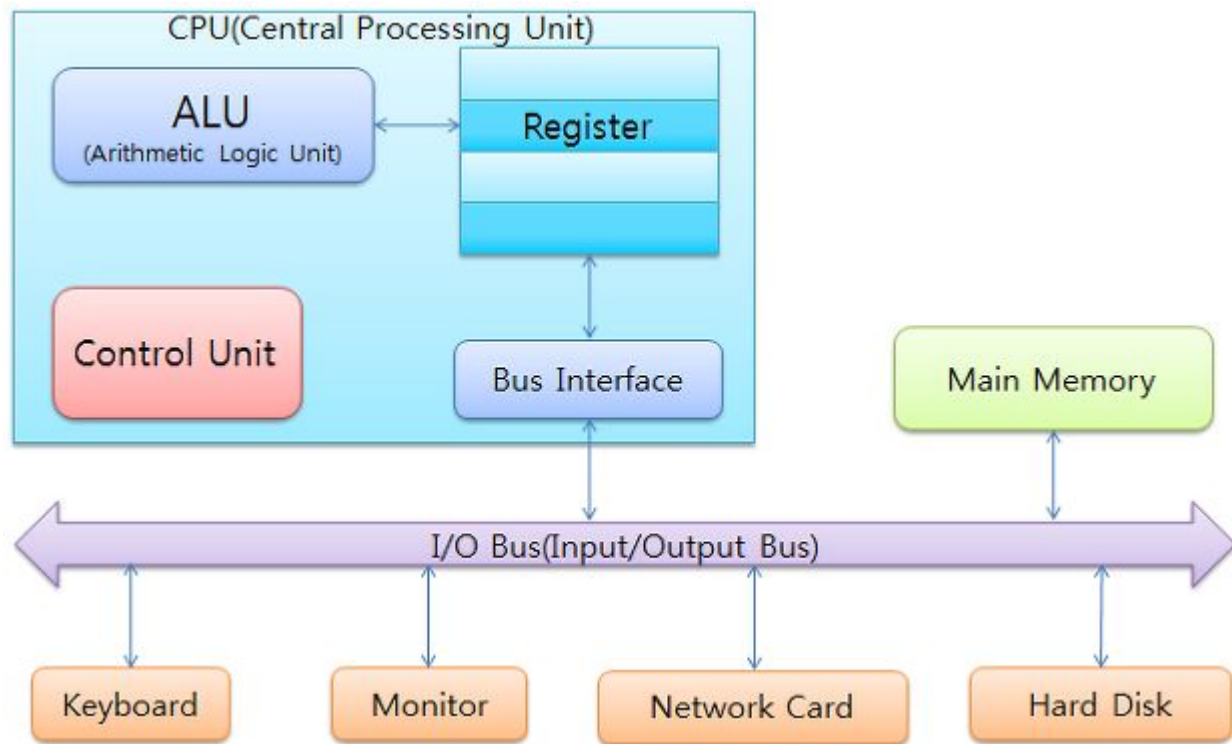
# Java

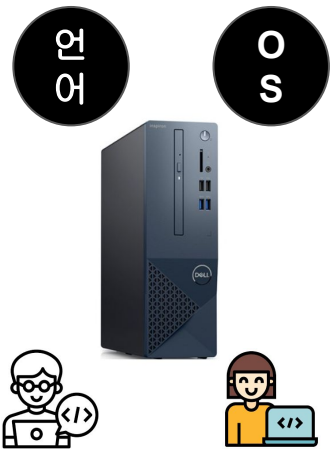


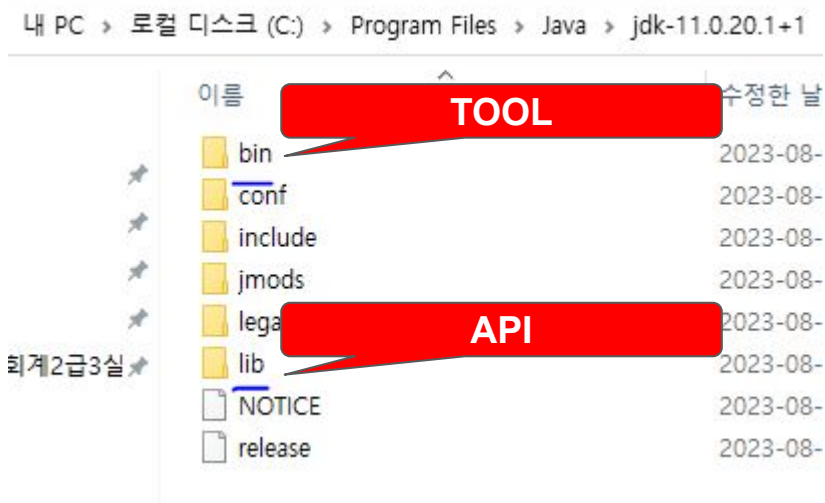
서버

인터넷

클라이언트







**JDK (자바개발환경)**  
**javac, javap**

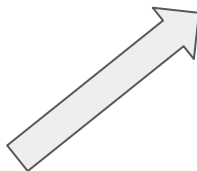
**JRE (자바실행환경)**  
**java, javaw, library(API)**

**JVM (자바가상머신)**  
**JIT 컴파일러**

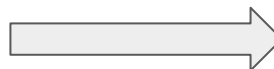
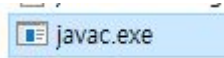


```
public class Hello {  
    public static void main(String[] args)  
    {  
        System.out.println("Hello Wolrd!");  
    }  
}
```

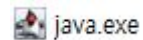
프로그래밍언어  
- 자바



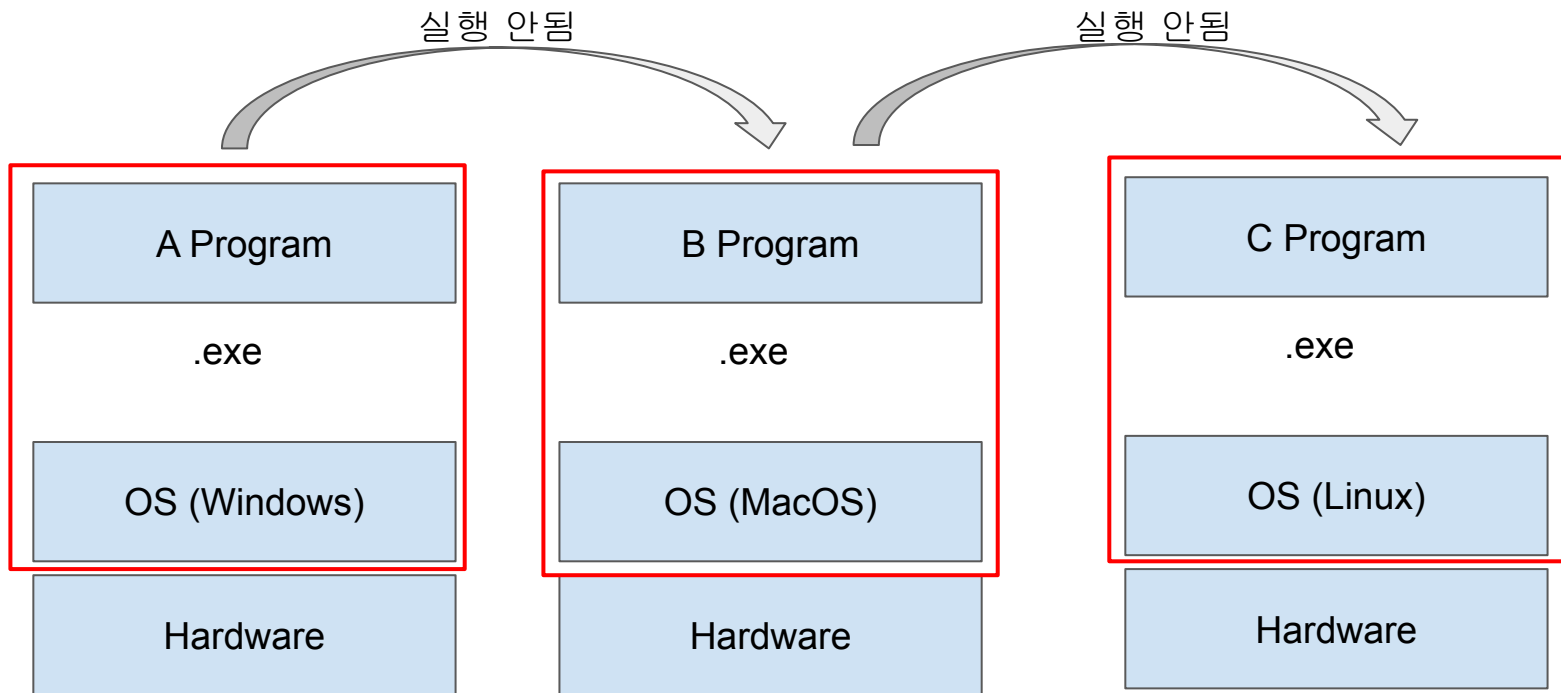
JDK에 포함된  
컴파일러



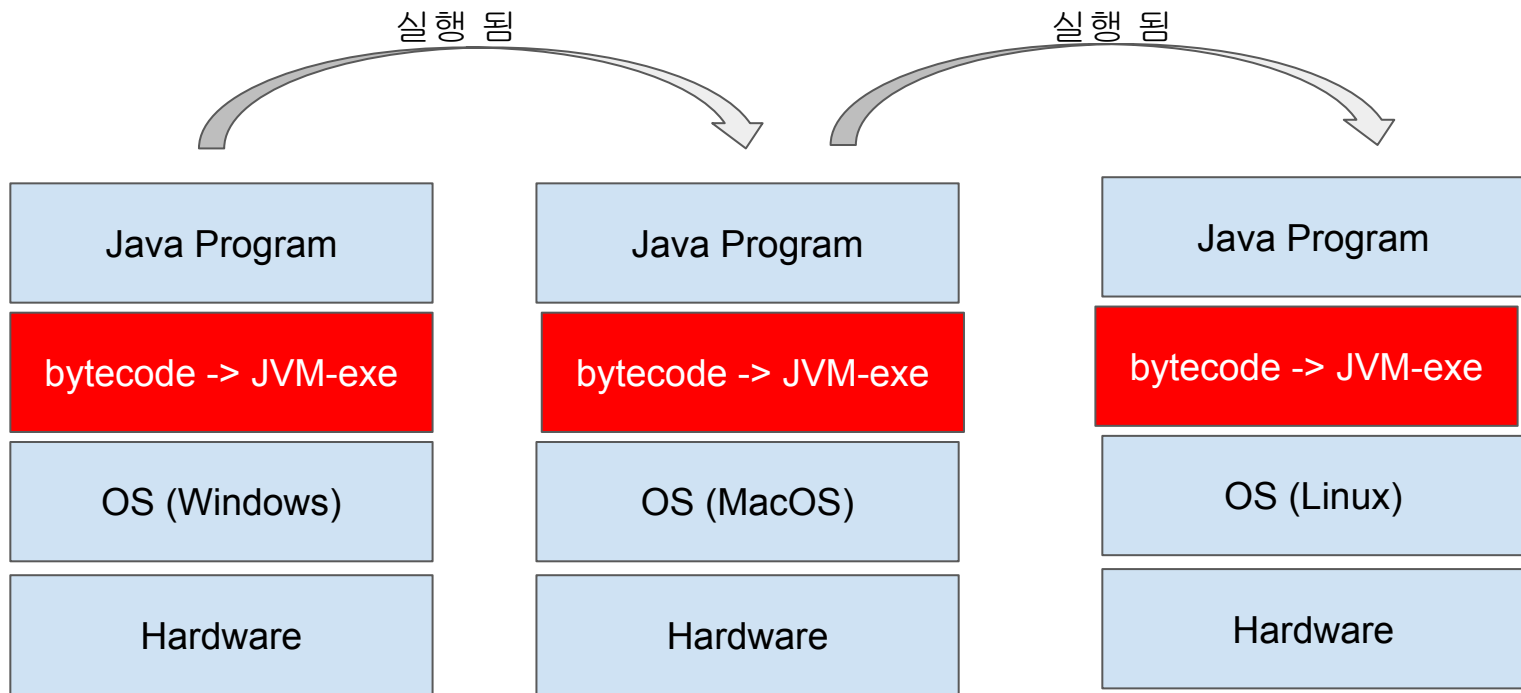
JDK에 포함된 실행



0010101111000011010  
0001101000000001111  
0000111100111100111



종속적





JDK에서 처리

Java Source  
File  
(.java)

컴파일  
(javac.exe)

Java Byte  
Code  
(.class)

Run을 하는 순간  
(실행하는 순간)  
java.exe

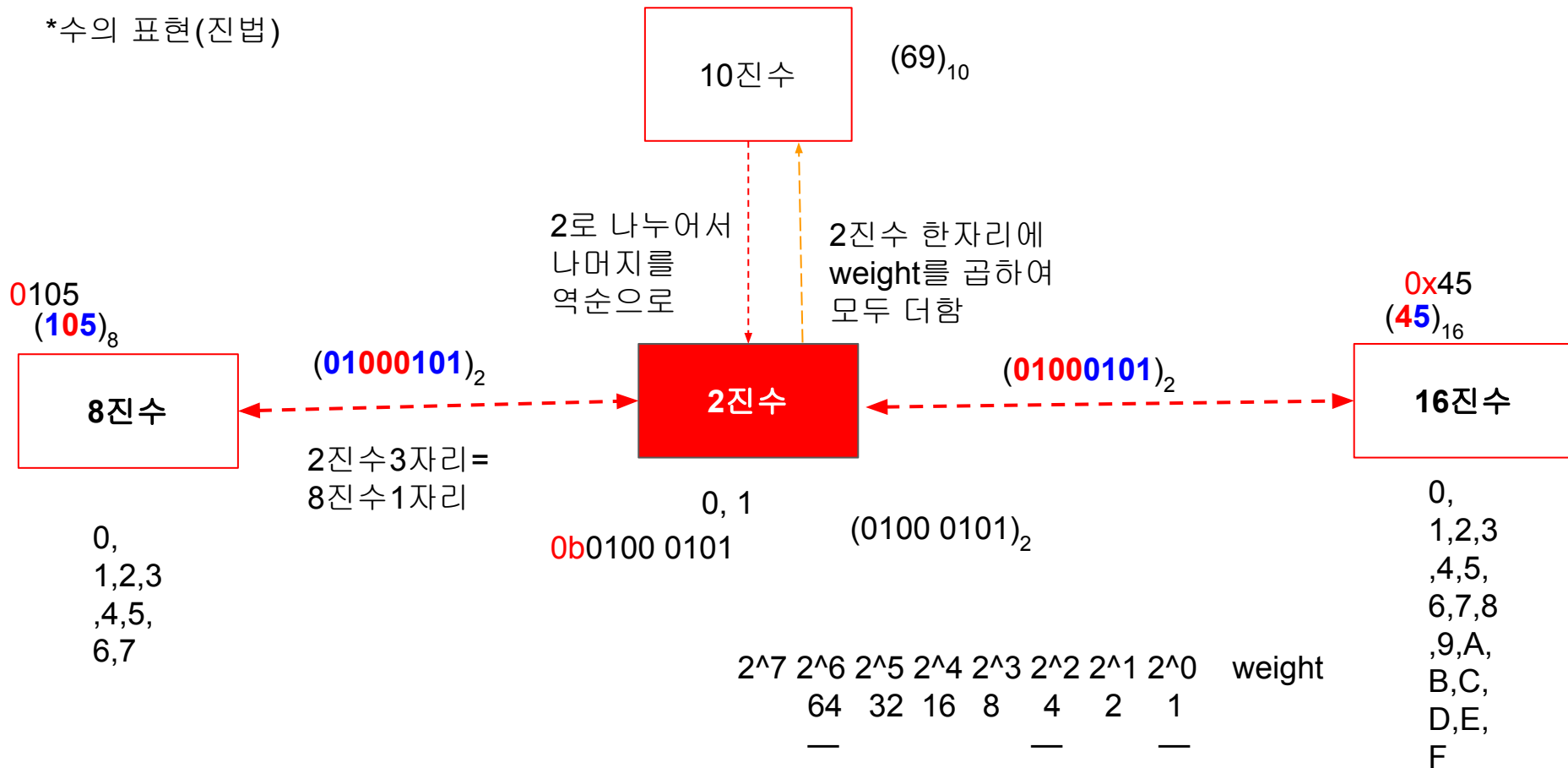
JVM이 동작 함

Class loader

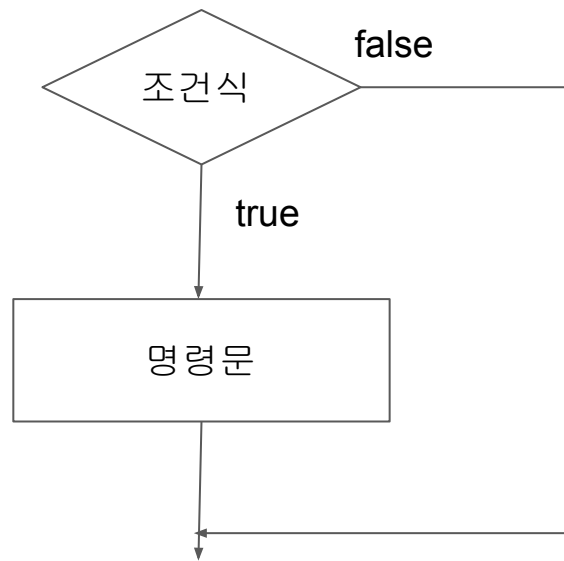
Execute

java.exe Hello (실행)  
⇒ 원격프로세서 호출 (JVM)

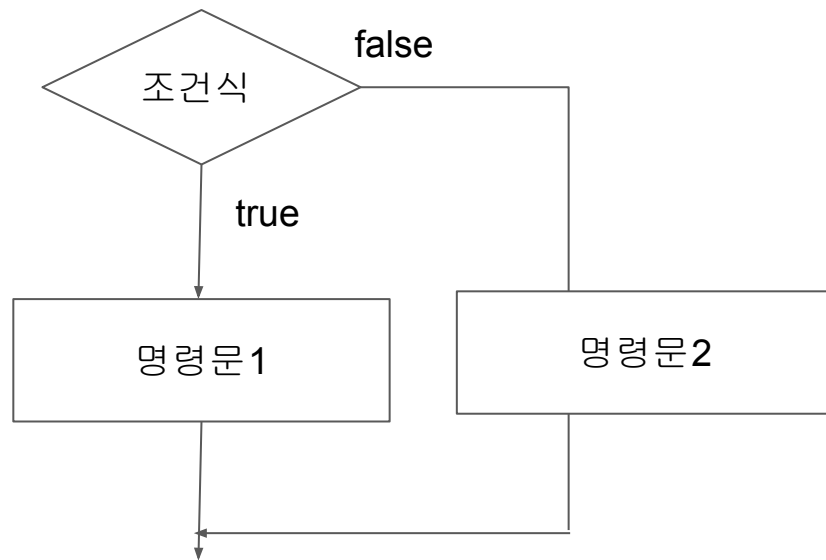
\*수의 표현(진법)

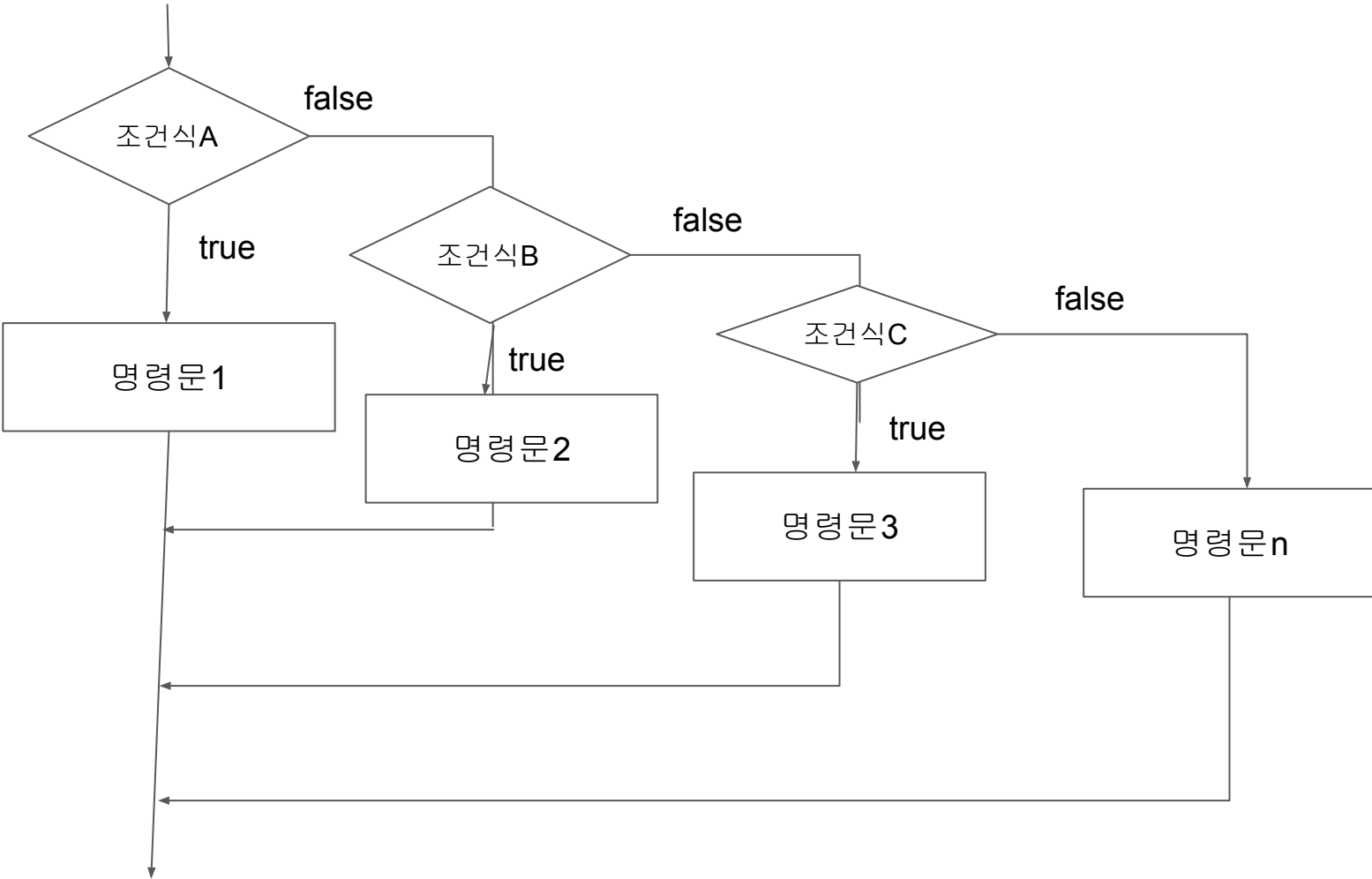


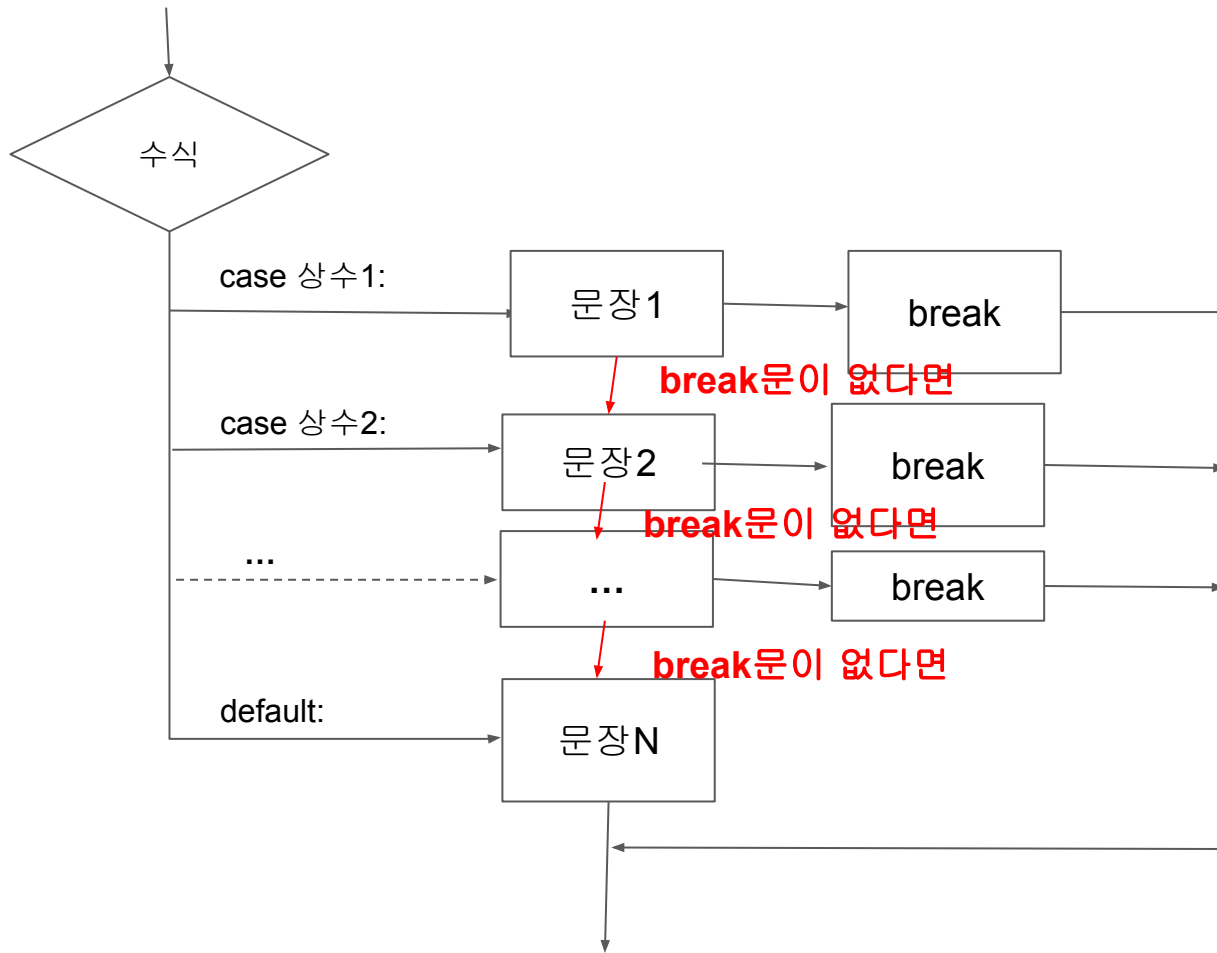
if문



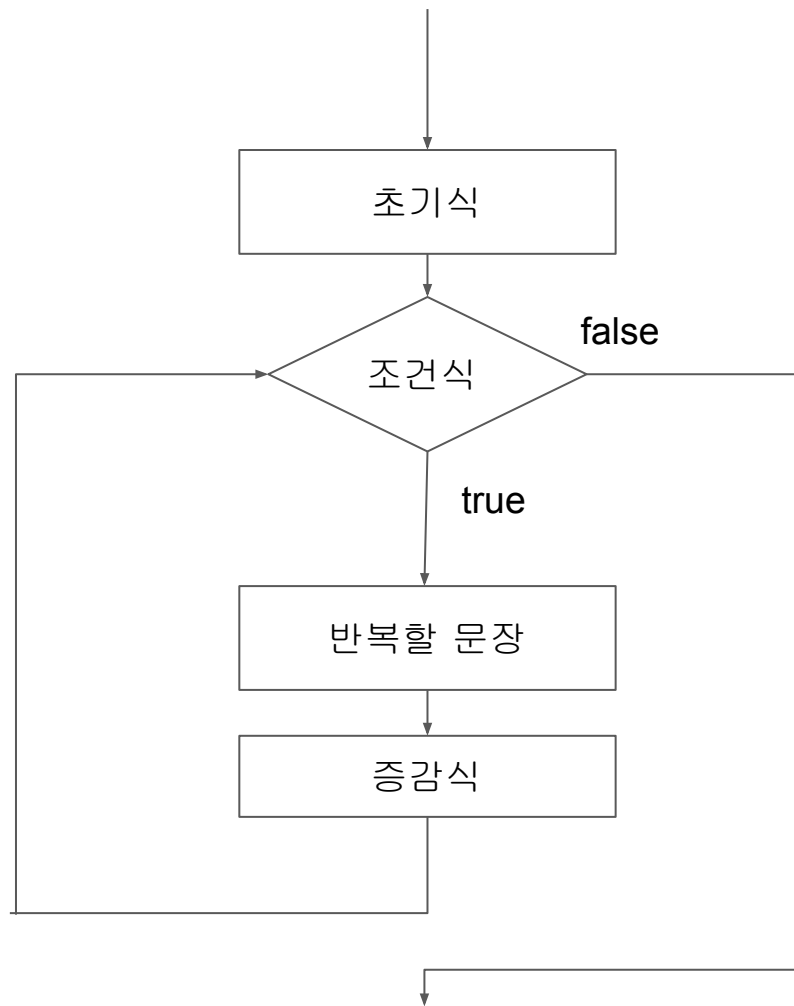
if ~ else 문



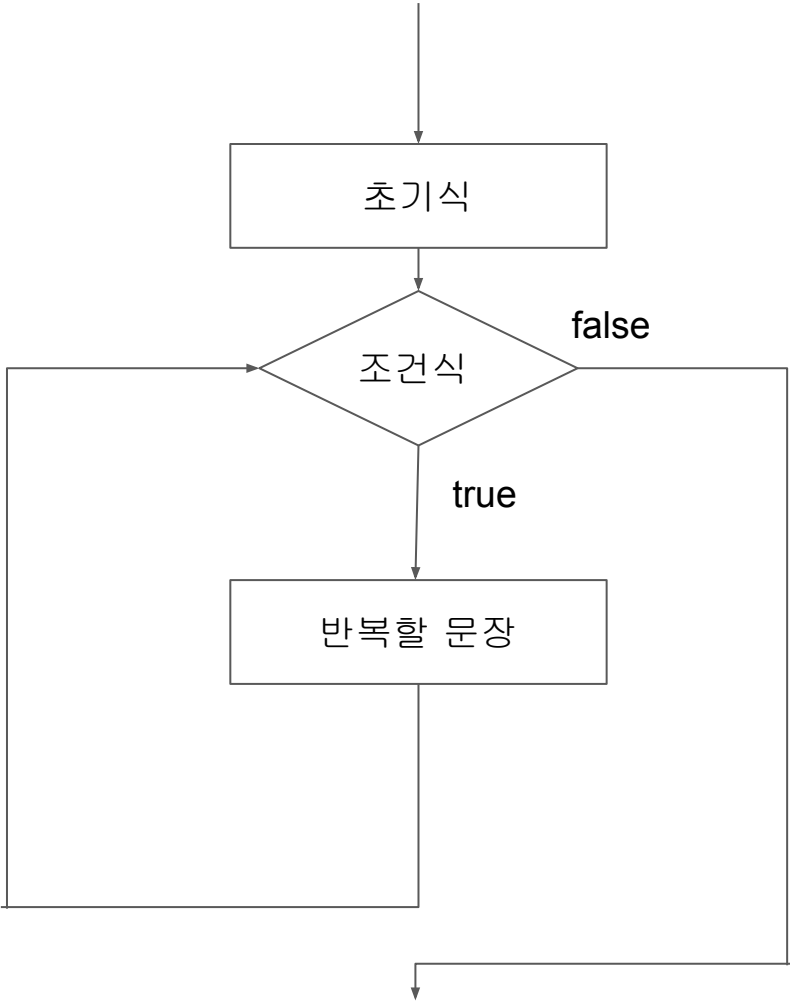




for문

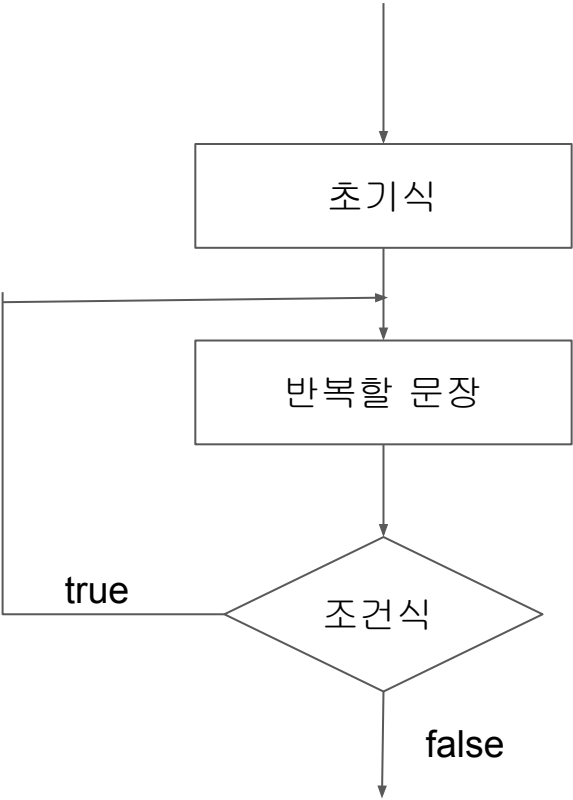


while문





do-while문



JDK에서 처리

Java Source  
File  
(.java)

컴파일  
(javac.exe)

Java Byte  
Code  
(.class)

Run을 하는 순간  
(실행하는 순간)  
java.exe

JVM이 동작 함

Class loader

Execute

java.exe Hello (실행)  
⇒ 원격프로세서 호출 (JVM)

JVM의 Memory model (Runtime Data Area)

Thread

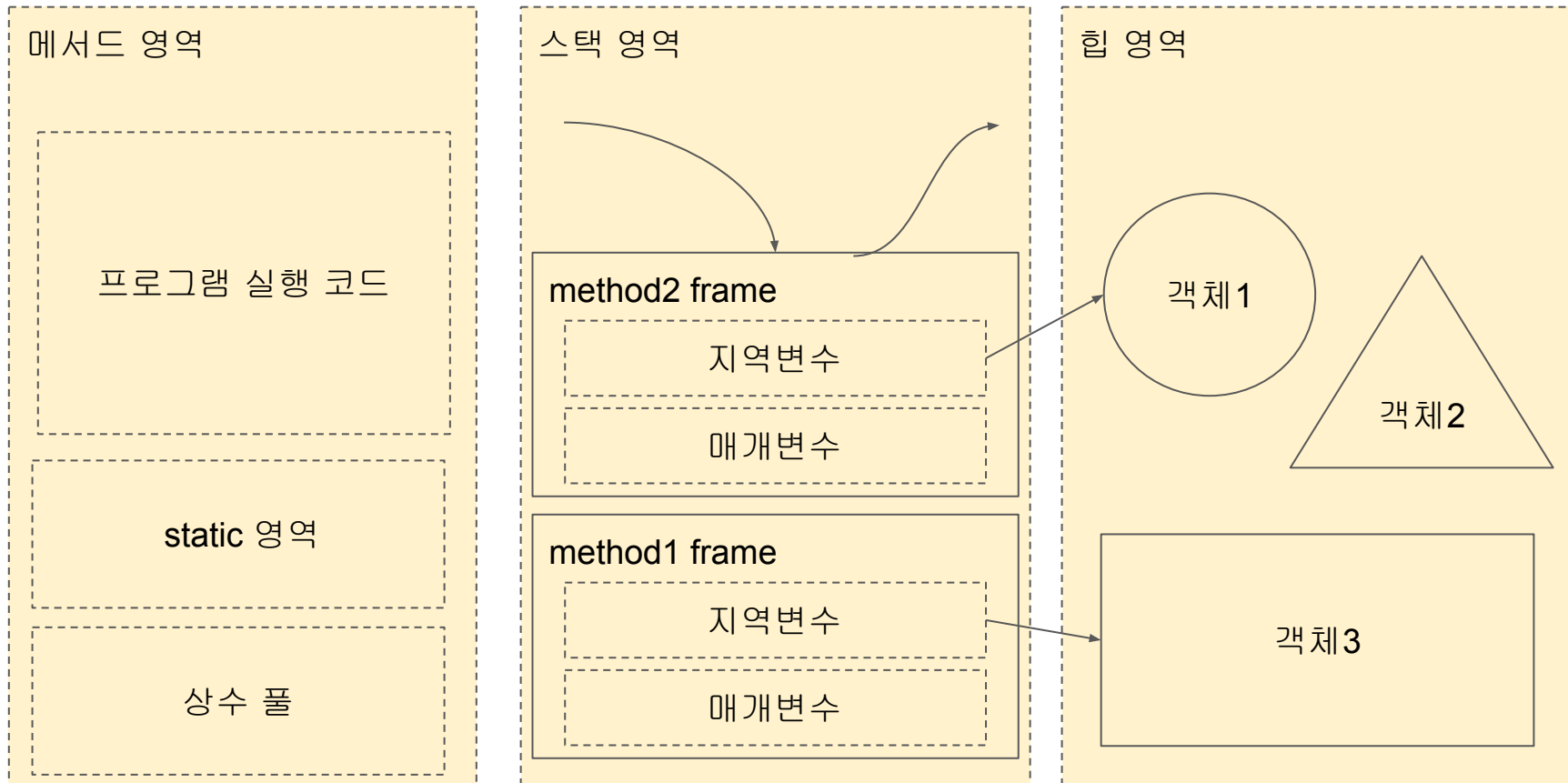
Method Area

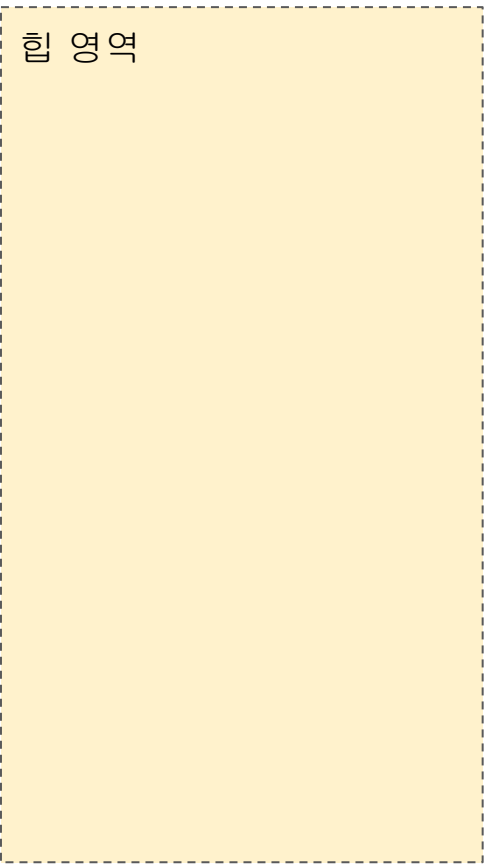
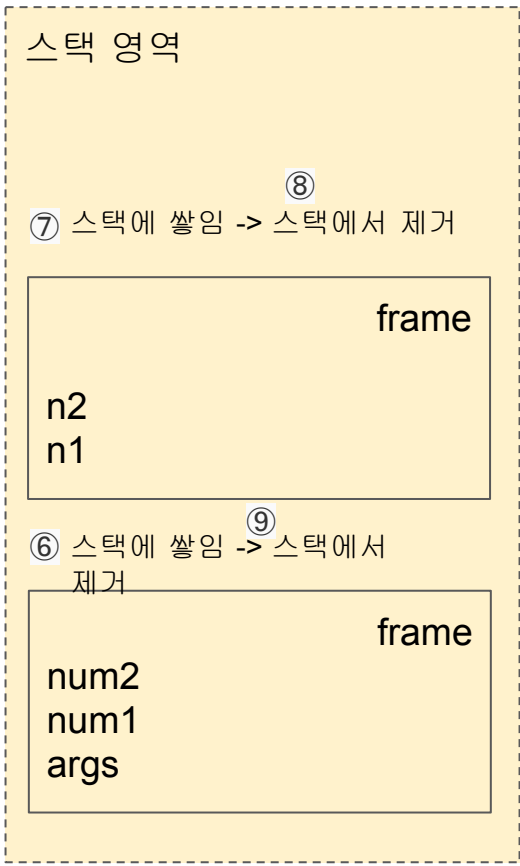
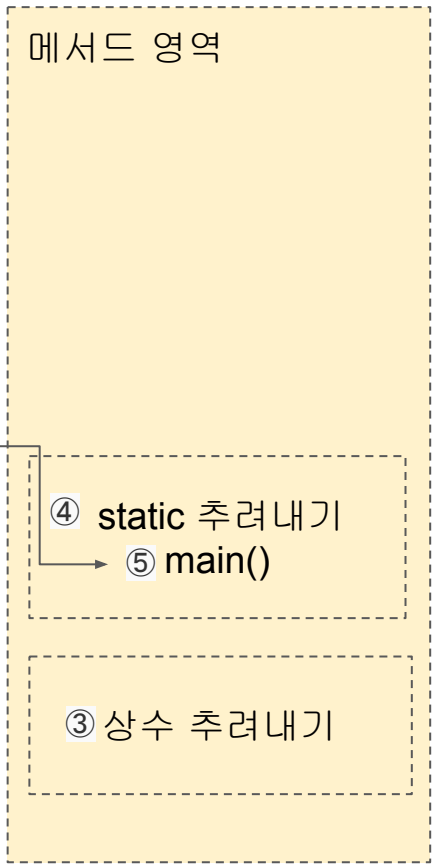
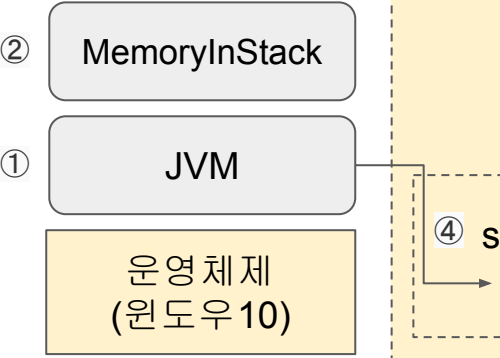
Heap Area

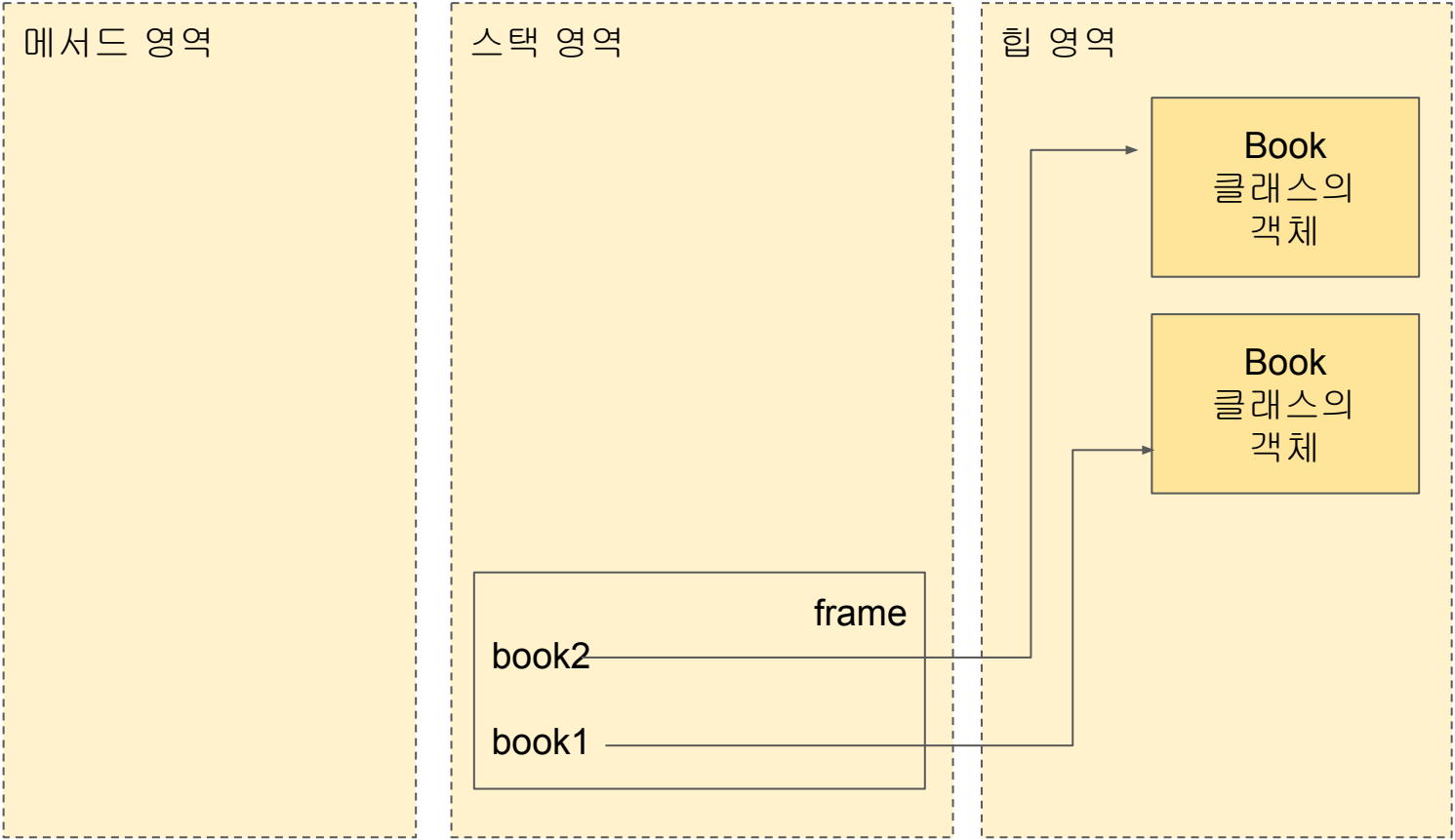
Stack Area  
(Call Stack Frame  
Area)

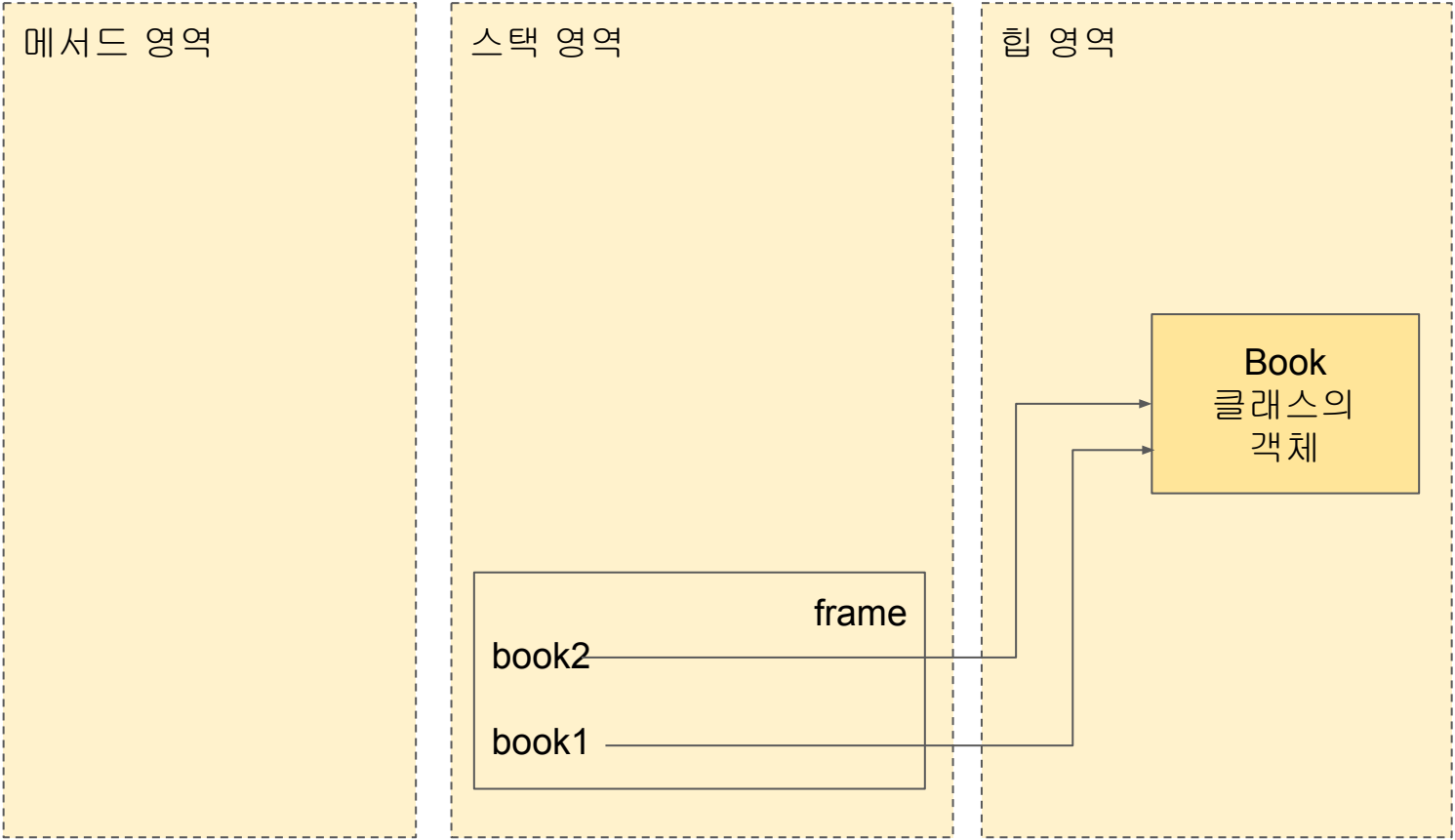
Runtime Constant  
Pool  
(Literal Pool)

## 런타임 data 영역들









기본자료형

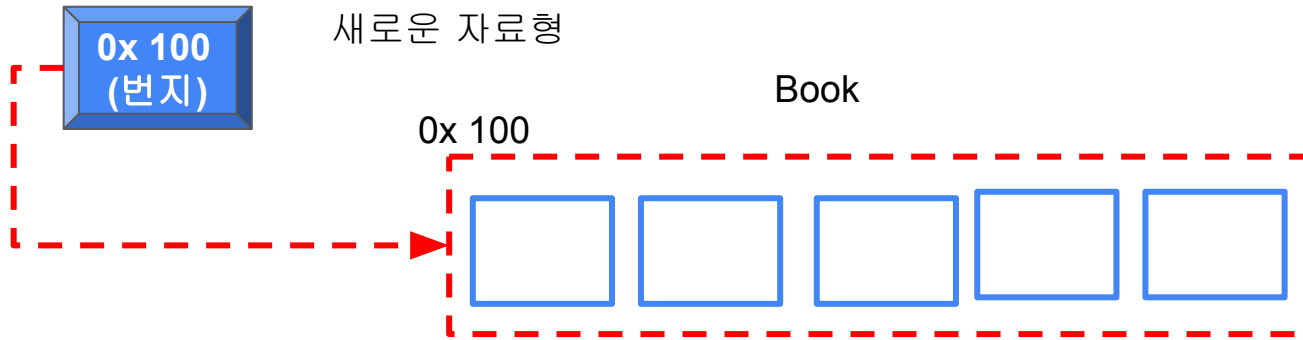


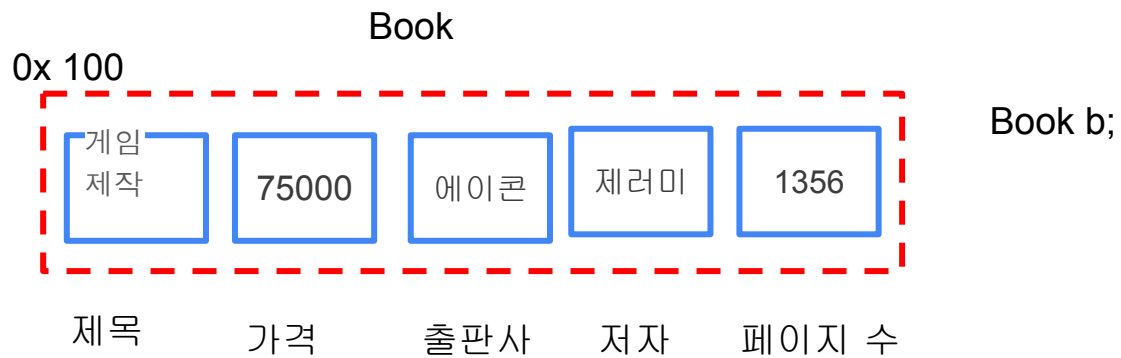
int a;

새로운 자료형

Book

Book b;

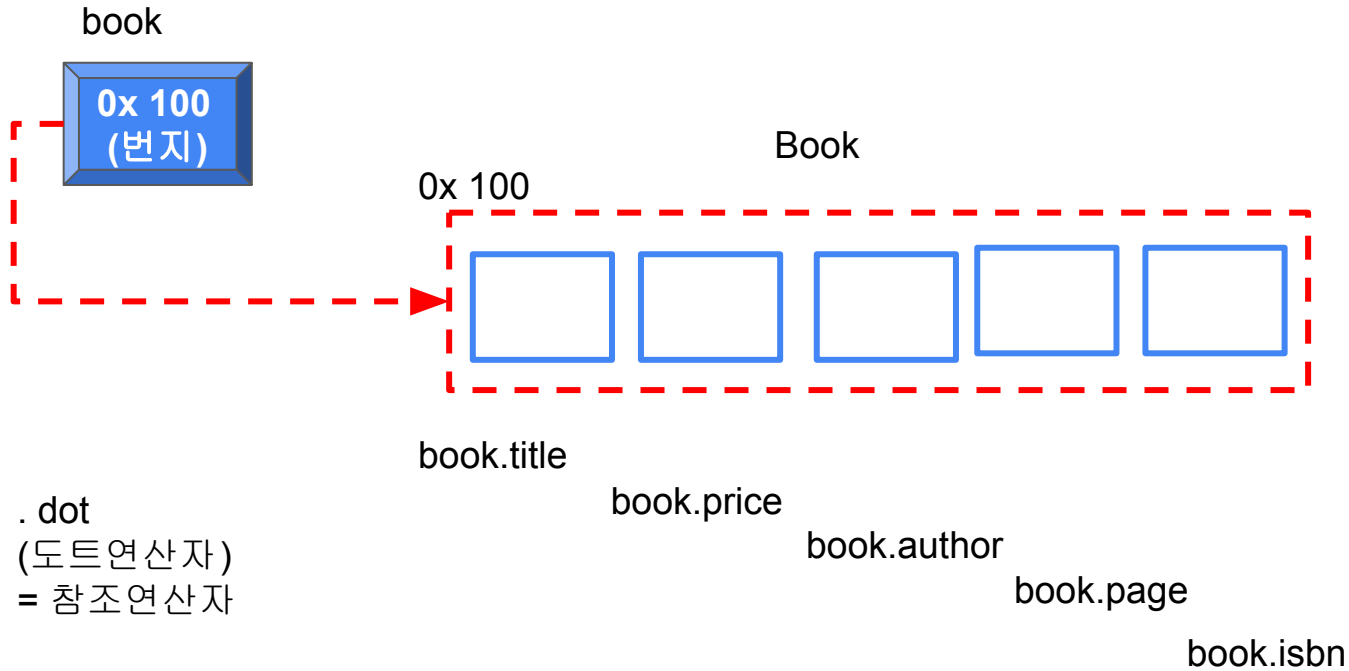




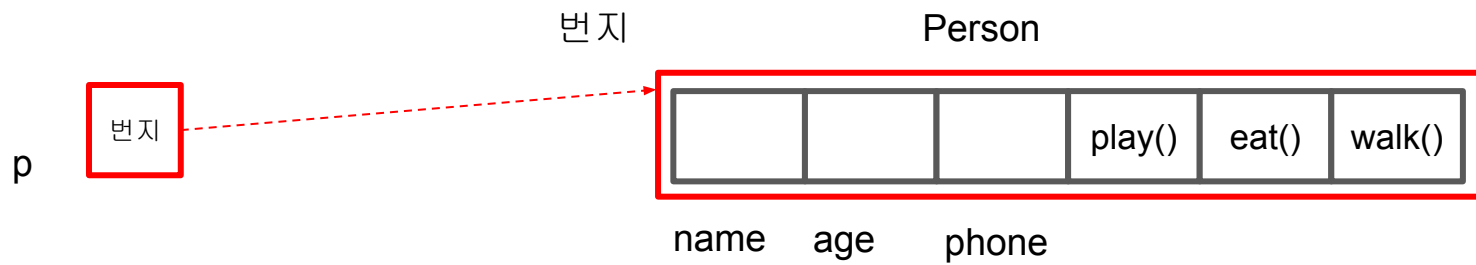
기억공간을 하나의 구조(객체)



```
Book book = new Book();
```

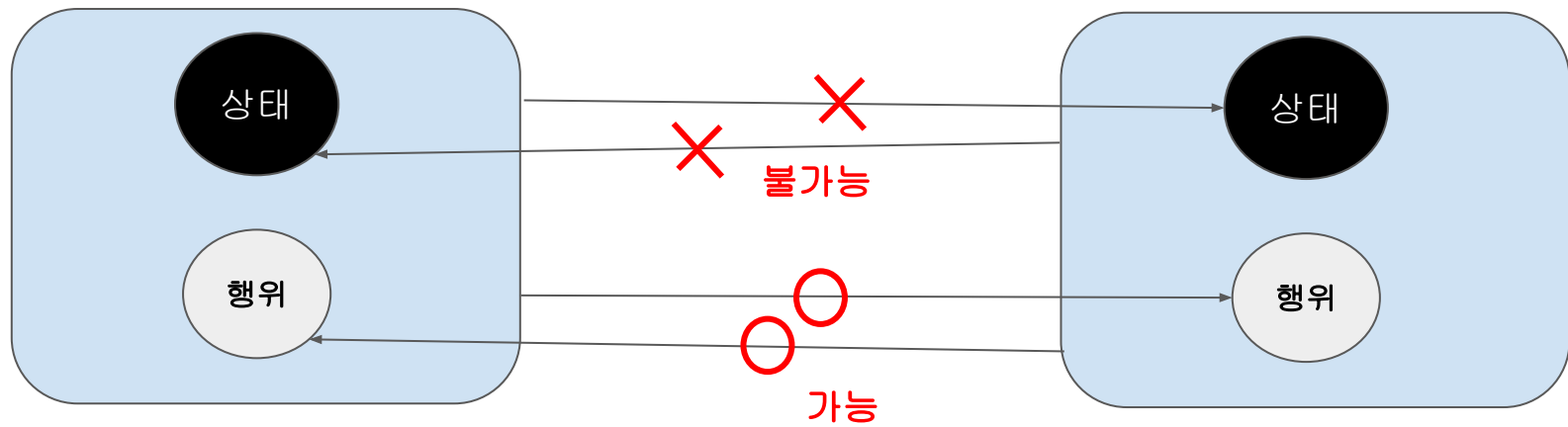


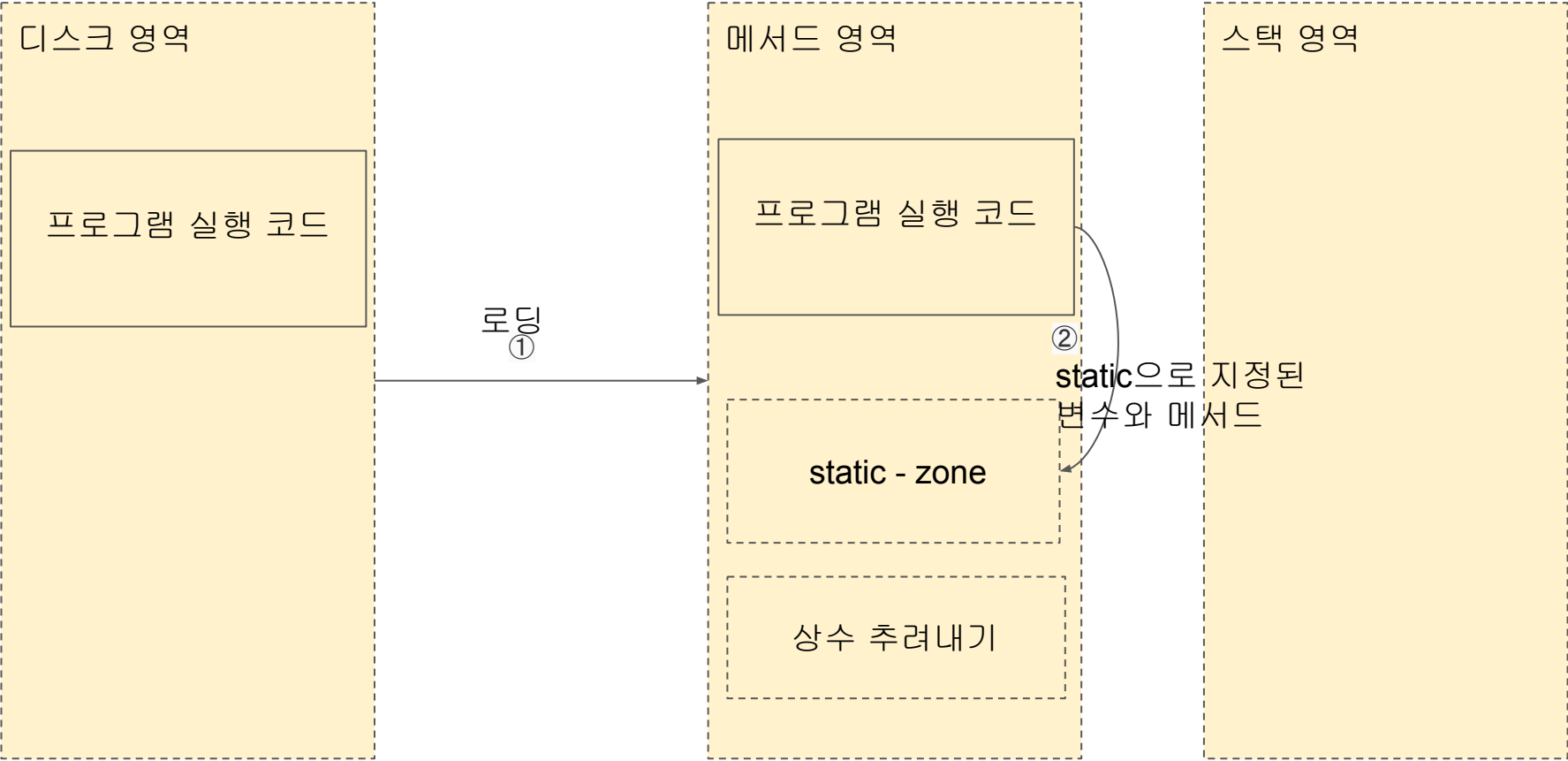
```
Person p = new Person();
```

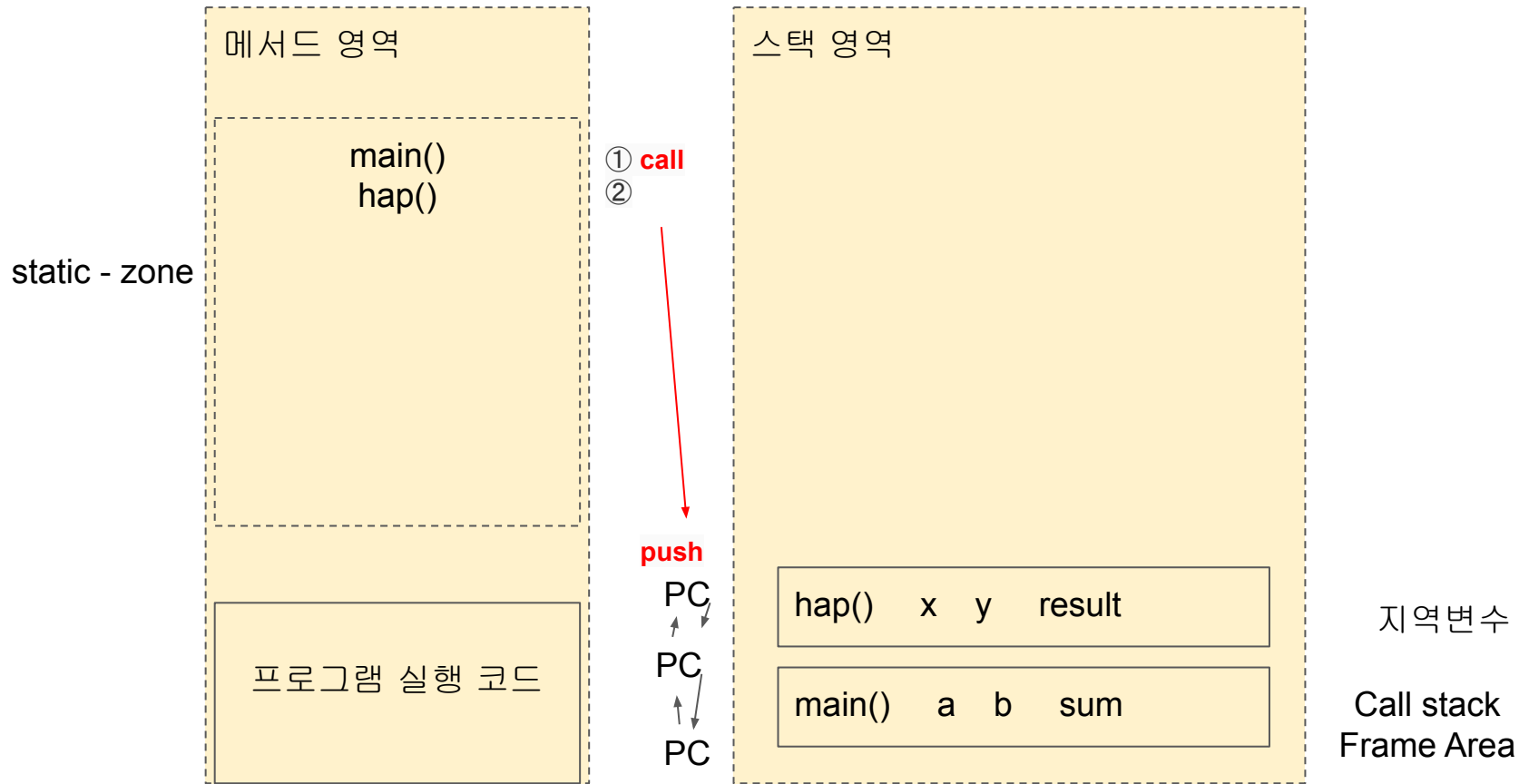


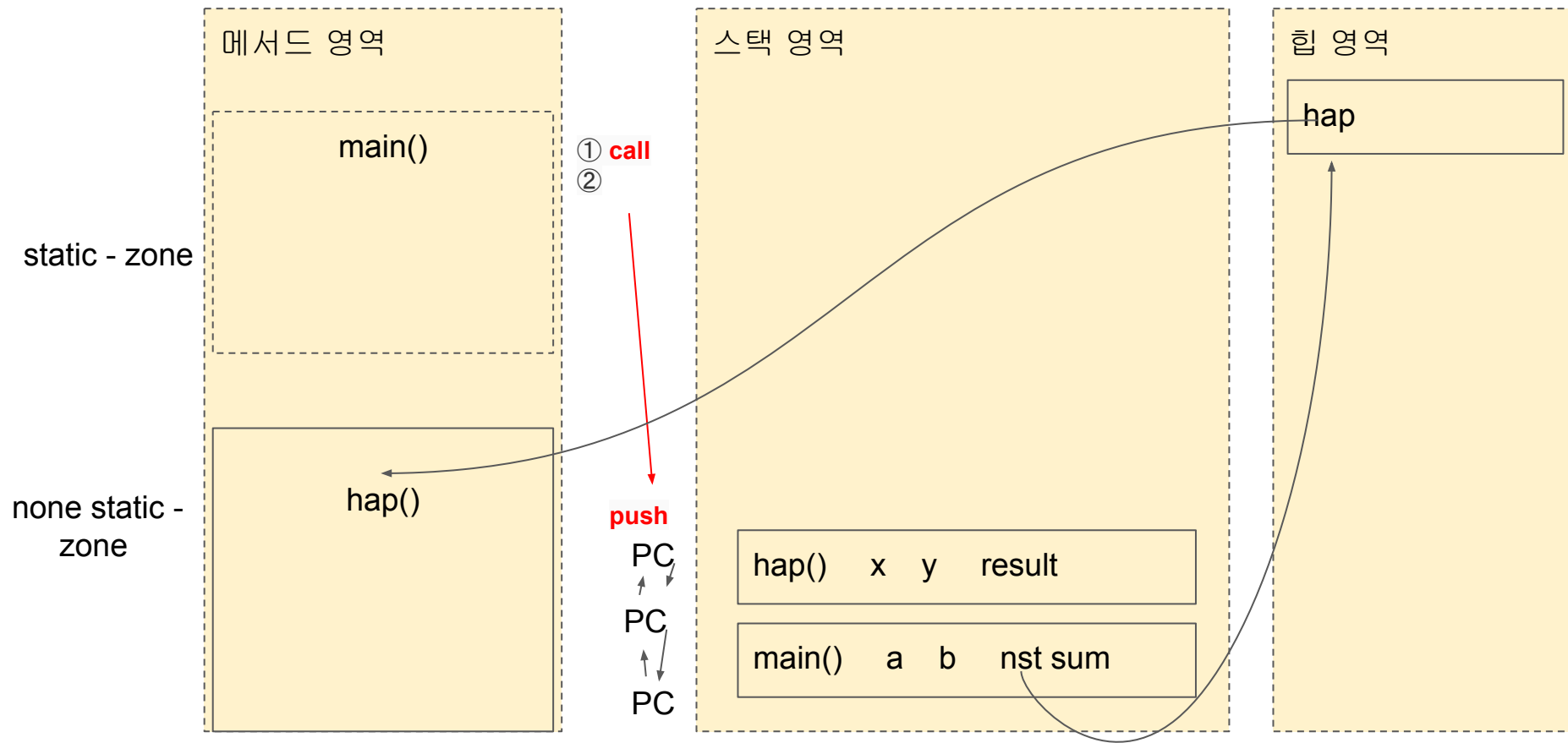
A class

B class









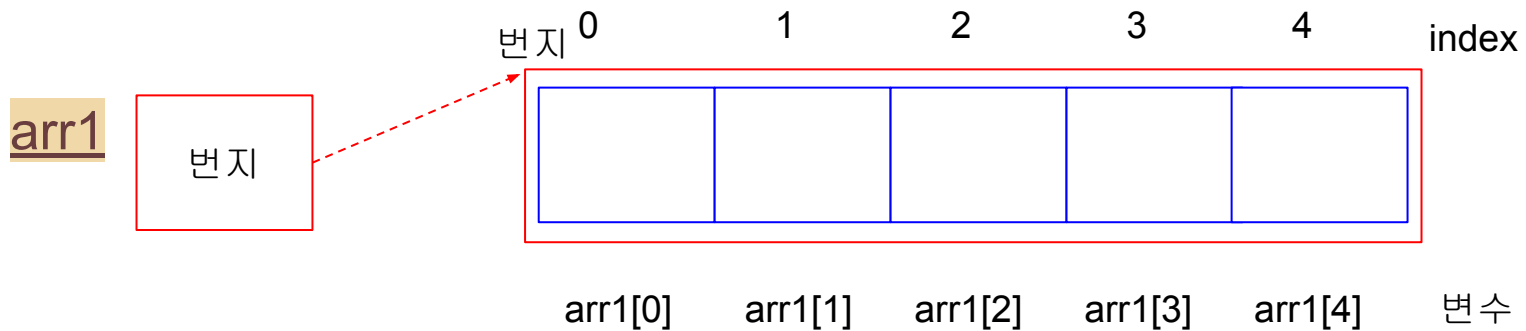
`int[ ]`      `int` 여러 개 자료형



1차원 배열 = [ ]

2차원 배열 = [ ][ ]

`new int[3];`





movies

번지

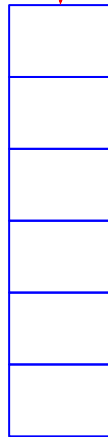
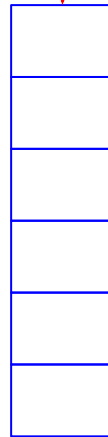
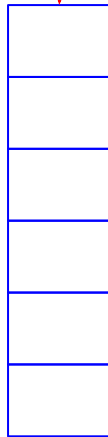
movies[0] movies[1] movies[2]  
번지

번지

번지

null

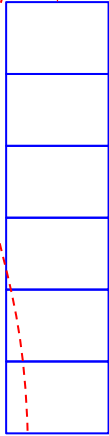
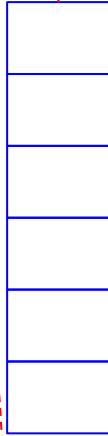
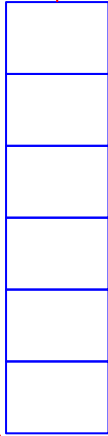
변수



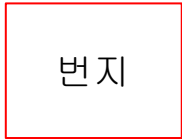
libray



Book[0] Book[1] Book[2]  
변지

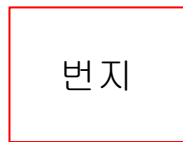


copylib



변수

movies



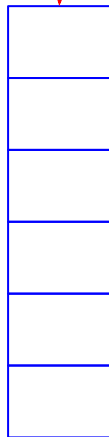
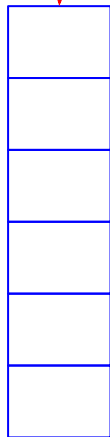
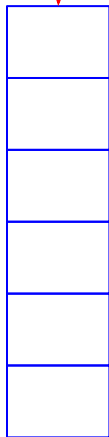
movies[0]  
번지



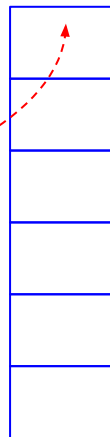
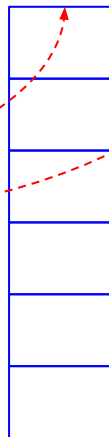
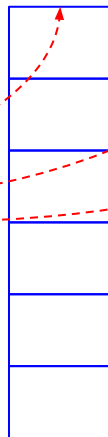
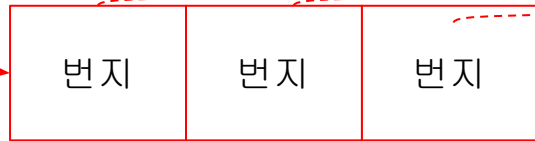
movies[1]

movies[2]

변수



copylib




arr[0][0]

arr[0][1]

arr[0][2]

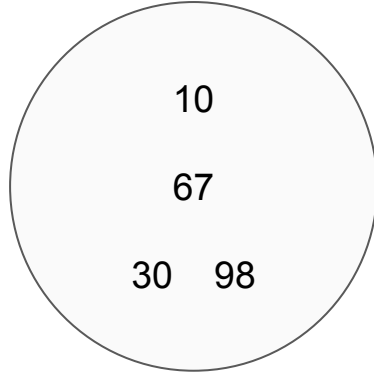
arr[1][0]

arr[1][1]

arr[1][2]

동일한 데이터 타입

정수 데이터



array

이질적인 데이터 타입

학생 데이터



class

### 일반 사원

이름,나이,  
전화번호,주소,  
입사일,근무부서  
부동산

### 관리 사원

이름,나이,  
전화번호,주소,  
입사일,관리부서  
부동산

### 비서

이름,나이,  
전화번호,주소,  
입사일,사무부서  
부동산

### 일용직

이름,나이,  
전화번호,주소,  
입사일,근로계약  
부동산

## 사원

이름,나이,  
전화번호,주소,  
입사일,부동소수점

### 일반 사원

사원의 정보를  
사용가능

### 관리 사원

사원의 정보를  
사용가능

관리부서

### 비서

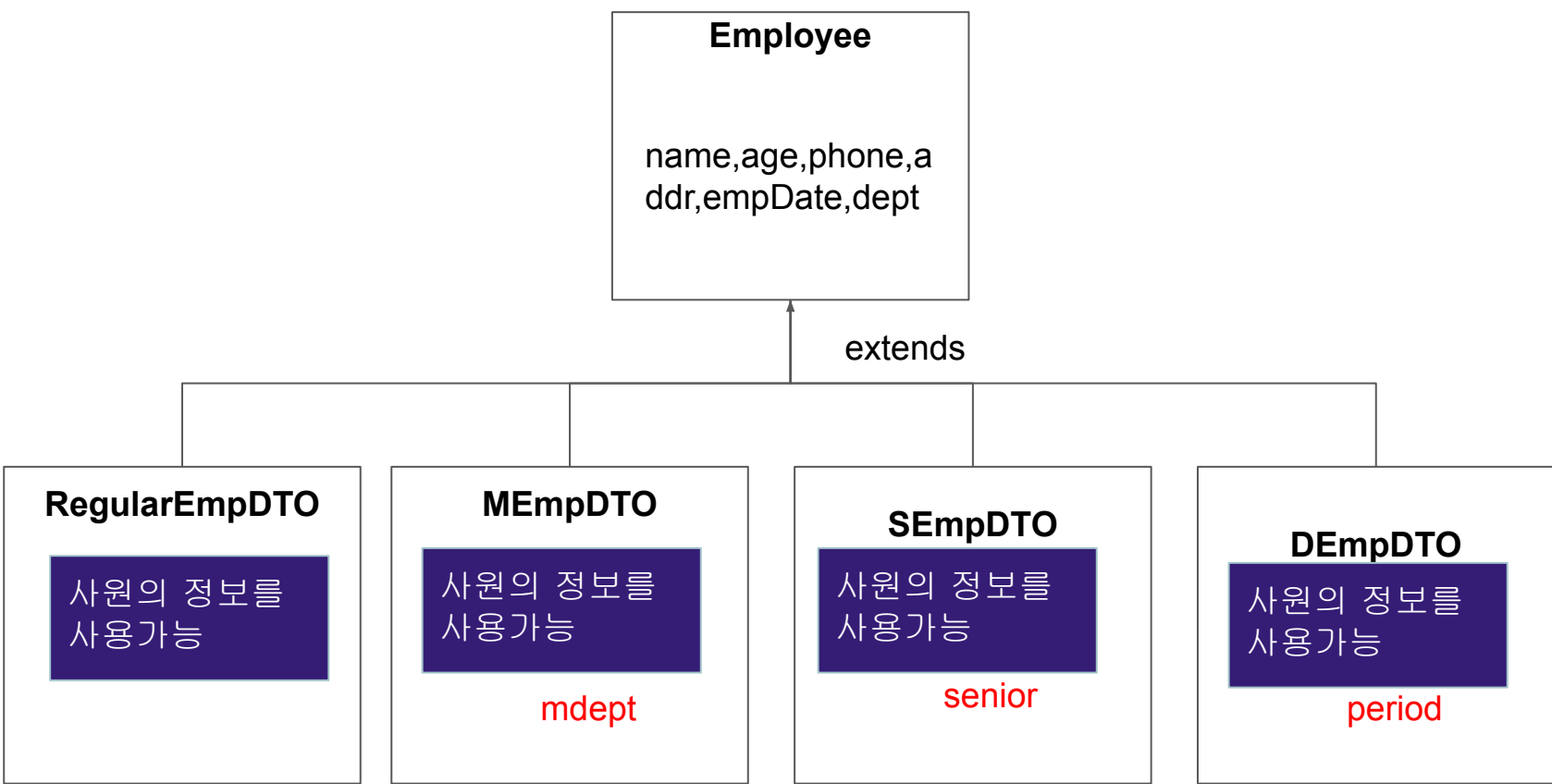
사원의 정보를  
사용가능

상급자

### 일용직

사원의 정보를  
사용가능

고용기간





부모

name,age,phone,a  
ddr,empDate,dept

extends : 부모가 자식들에게 자신의  
것을사용하라고 허락해 주는 것

자식A

사원의 정보를  
사용가능

자식B

사원의 정보를  
사용가능

mdept

자식C

사원의 정보를  
사용가능

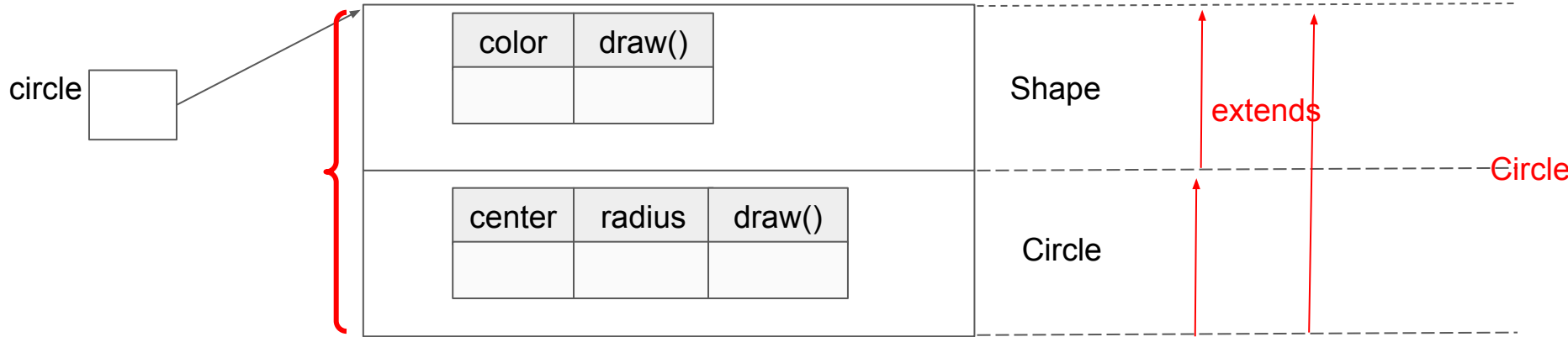
senior

자식D

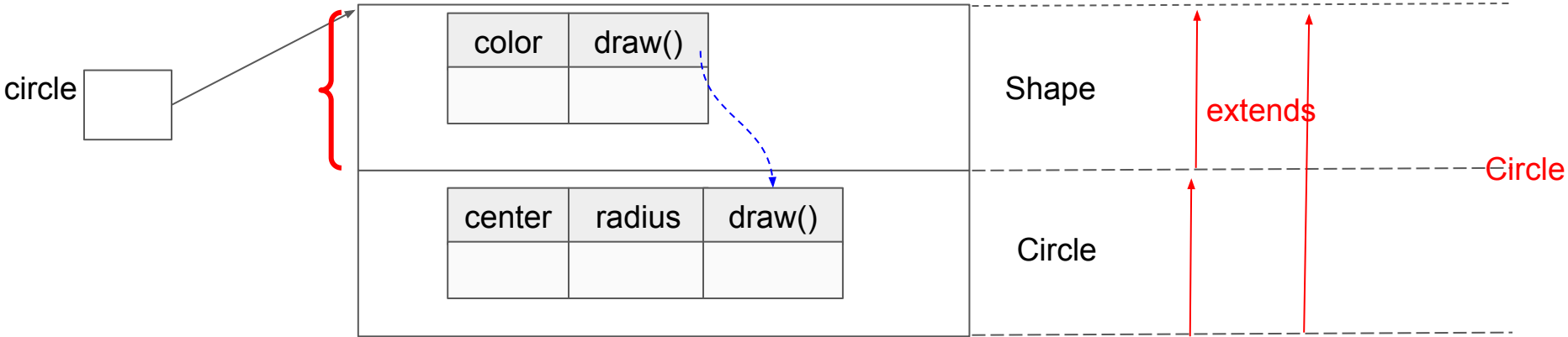
사원의 정보를  
사용가능

period

```
Circle circle = new Circle();
```



```
Shape circle = new Circle();
```

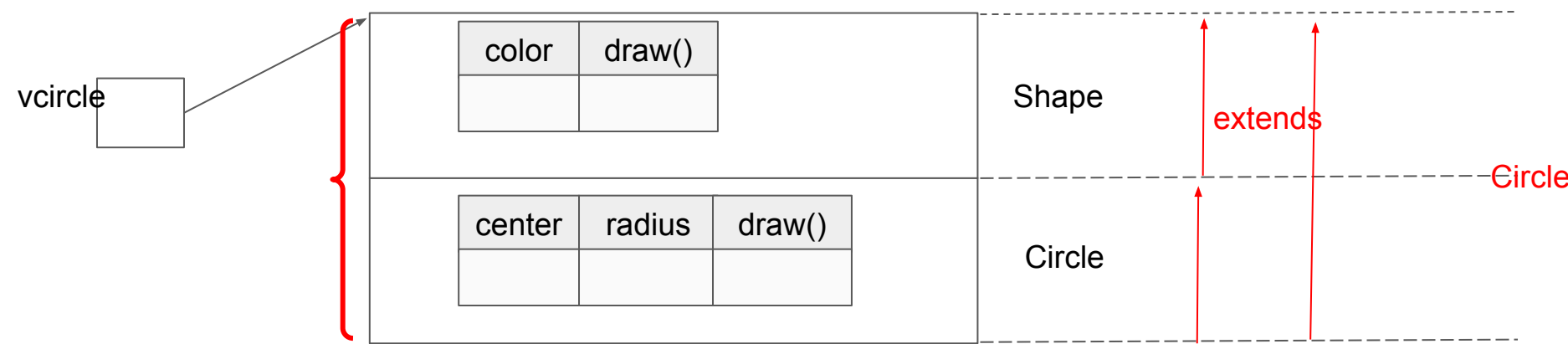


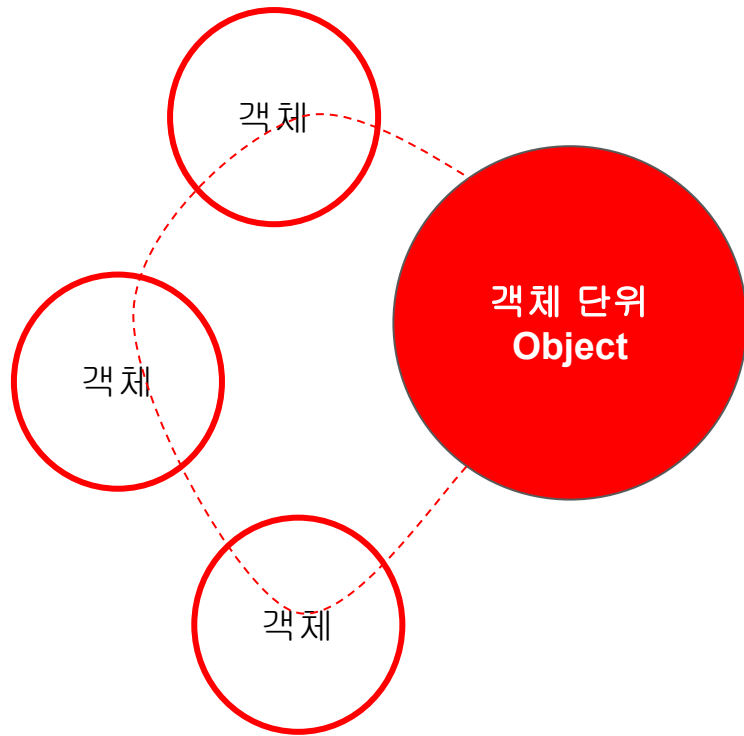
```
Shape circle = new Circle();
```

upcasting

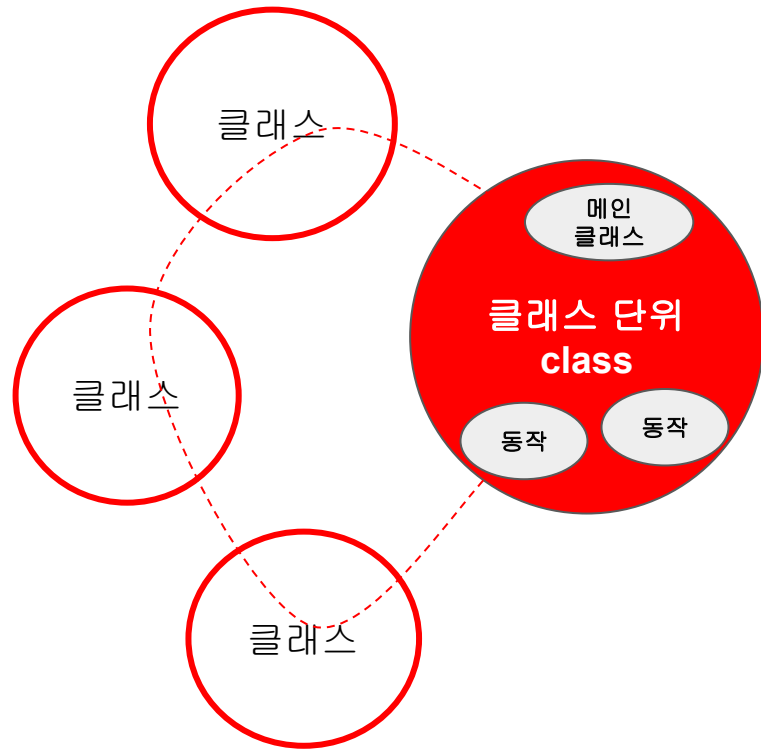
```
Circle vCircle = (Circle) circle ;
```

downcasting



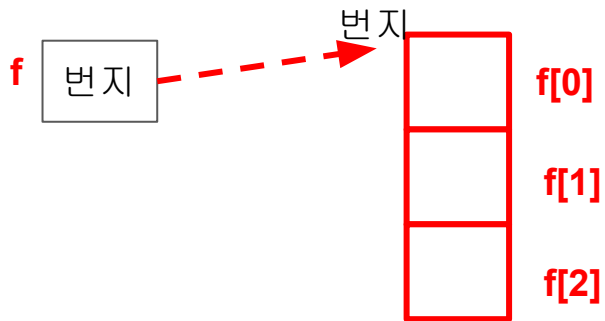


## 객체지향 프로그래밍



## 기본배열

```
float[ ] f = new float[3];
```



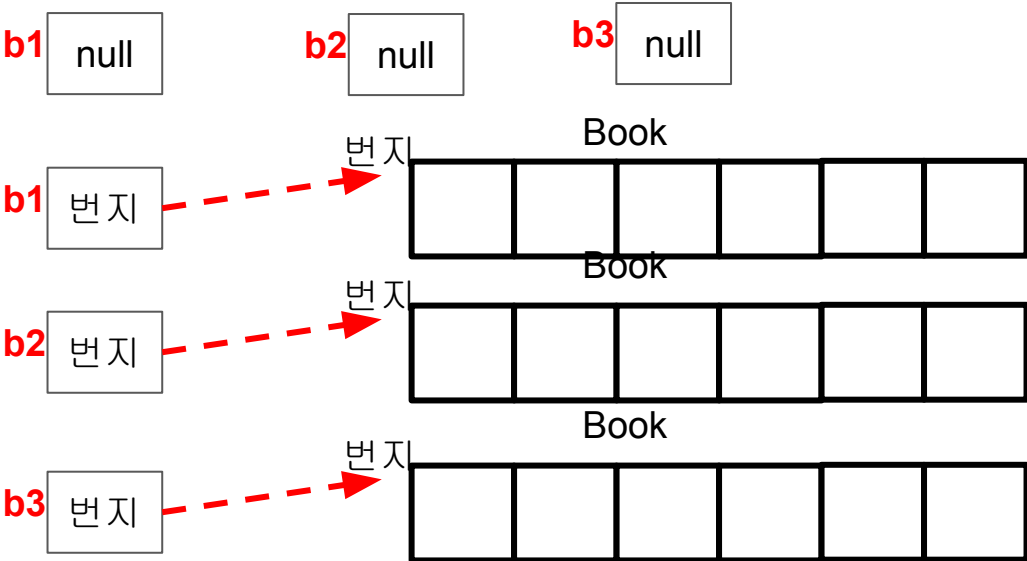
객체

```
Book b1, b2, b3;
```

```
b1 = new Book();
```

```
b2 = new Book();
```

```
b3 = new Book();
```



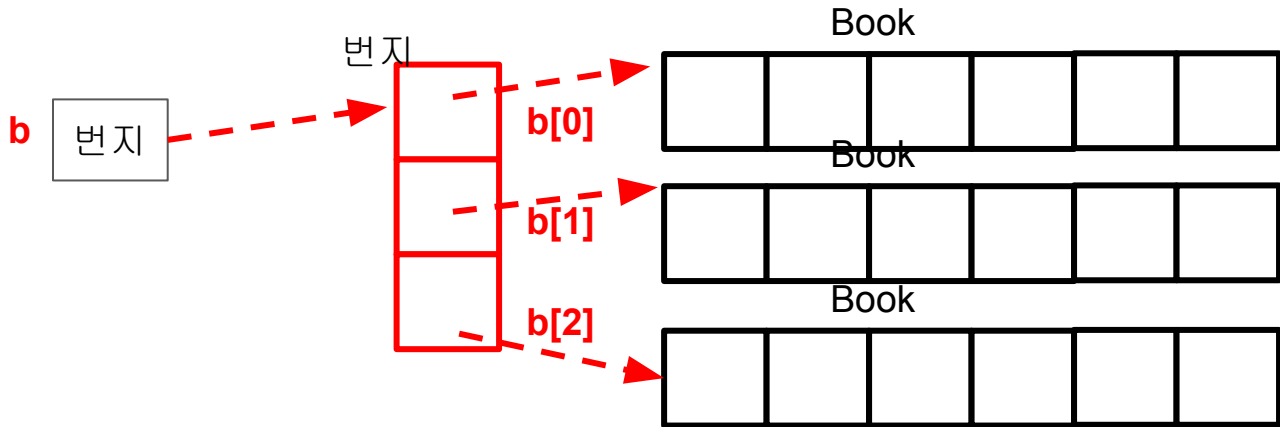
객체배열

```
Book[ ] b = new Book[3];
```

```
b[0] = new Book();
```

```
b[1] = new Book();
```

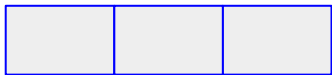
```
b[2] = new Book();
```



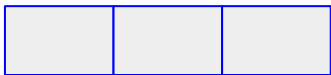
## 2차원 배열

1차원 배열을 3개 생성

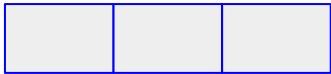
```
int[ ] kor = new int[3];
```



```
int[ ] eng = new int[3];
```

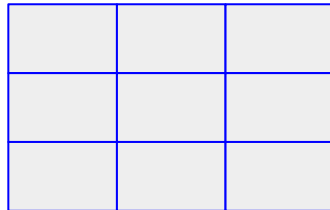


```
int[ ] mat = new int[3];
```



2차원 배열을 1개 생성 (이미지)

```
int[ ][ ] score = new int[3][3];
```





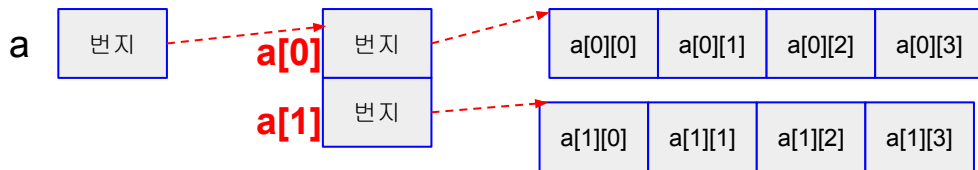
## 2차원 배열 생성

```
int[ ][ ] a = new int[행][열];  
int[ ][ ] a = new int[2][4];
```

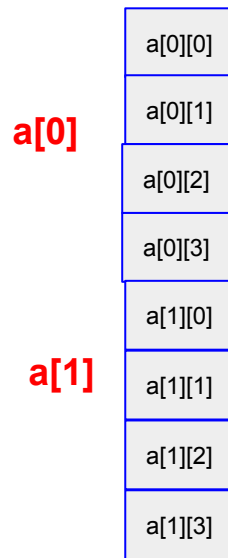
4열

2행

a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]

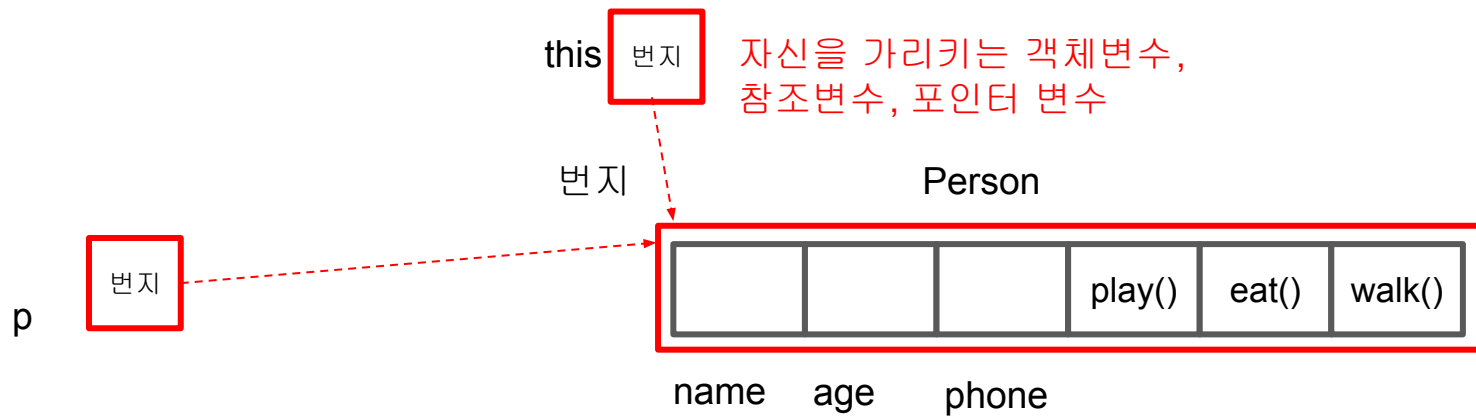


2차원 배열도 연속적인  
자료구조

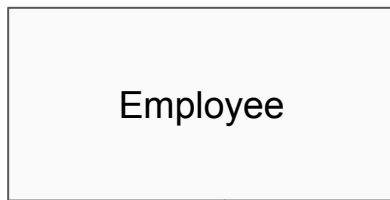


```
Person p = new Person();
```

생성자메서드



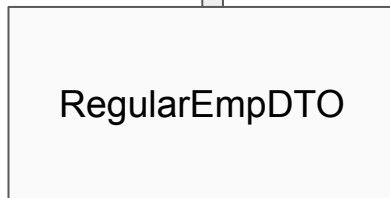
super class  
상위 클래스  
부모 클래스



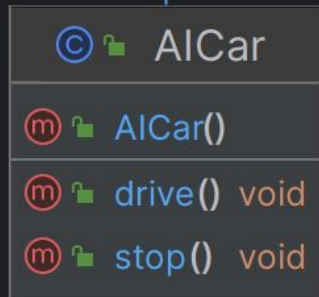
일반화  
추상화  
개념화  
포괄적

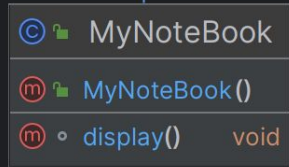
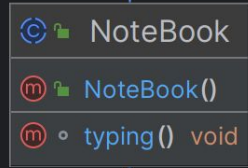
**extends** 상속, 확장

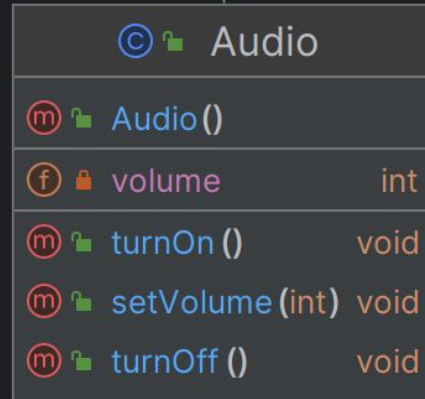
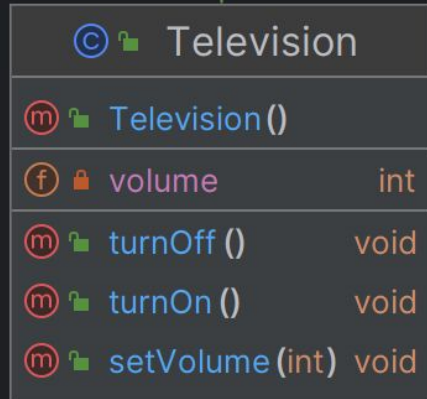
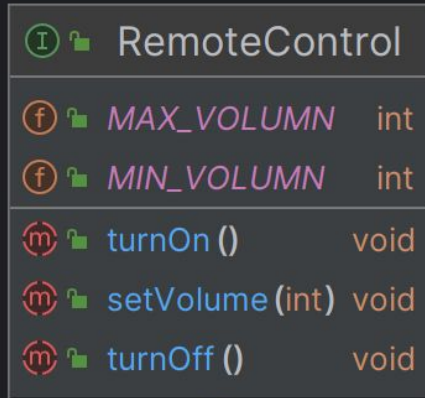
sub class  
하위 클래스  
자식 클래스  
파생 클래스

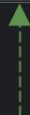
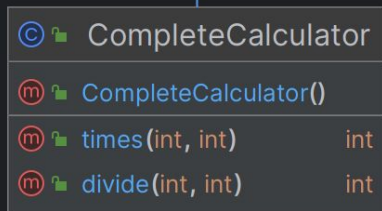
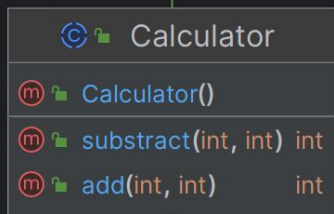
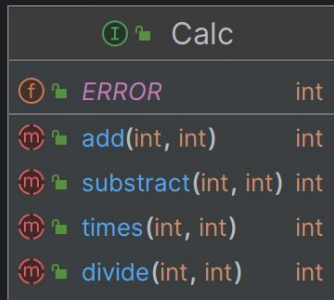


구체화  
세분화

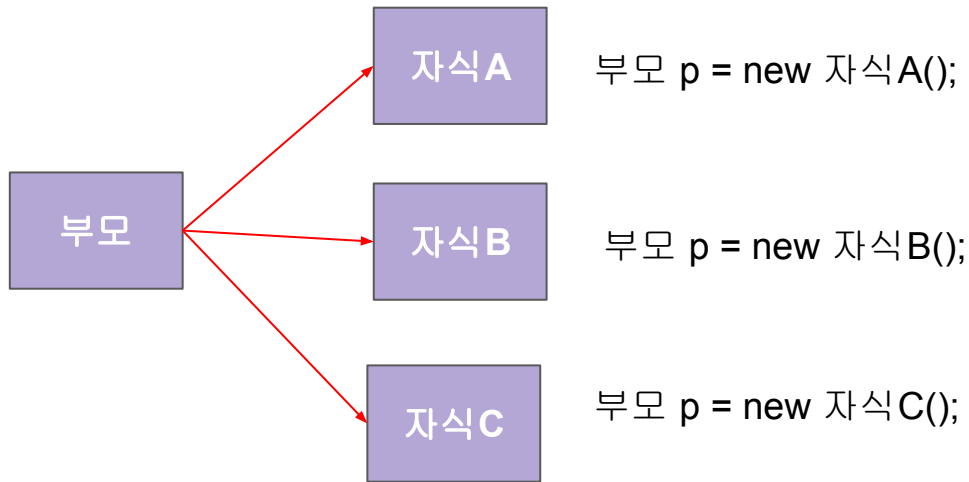




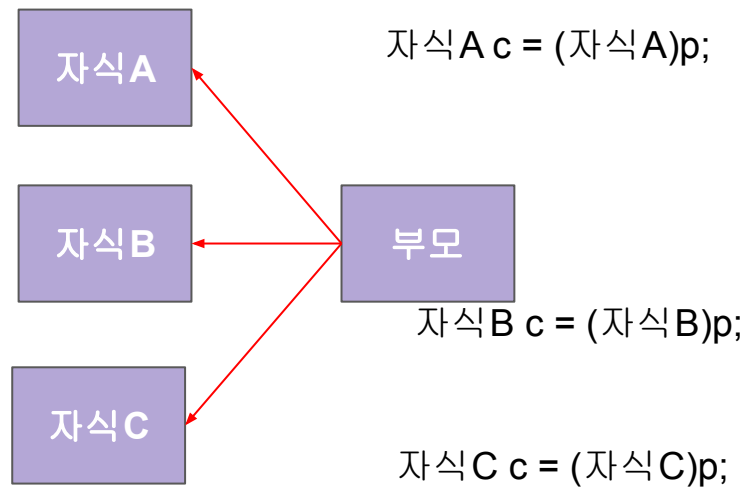




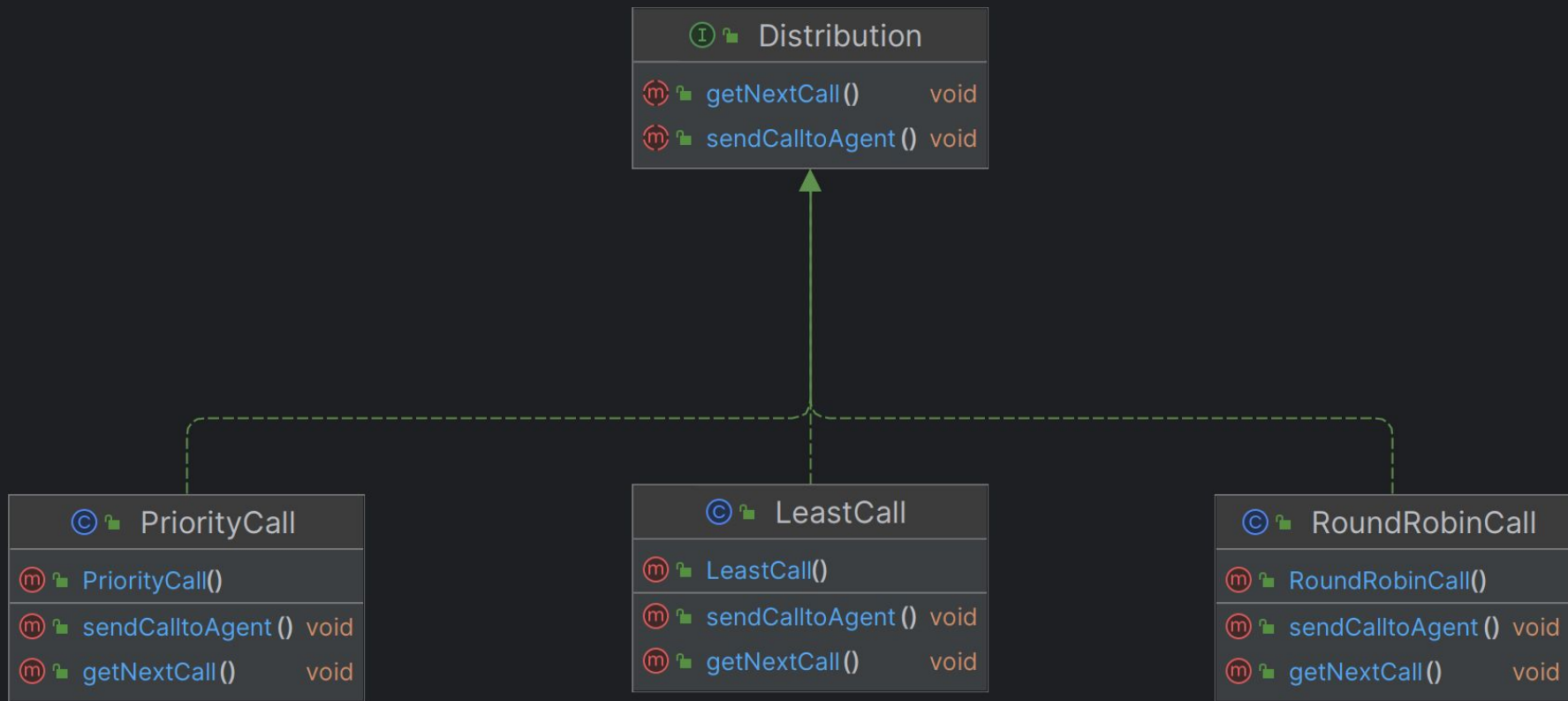
## Upcasting

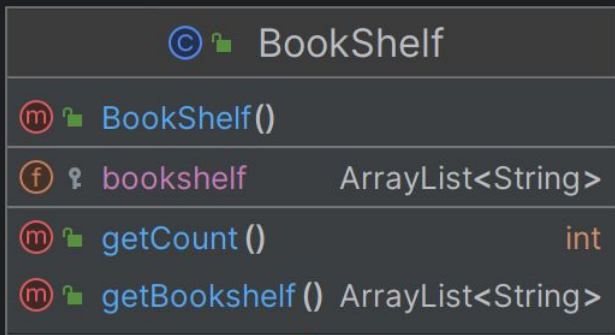
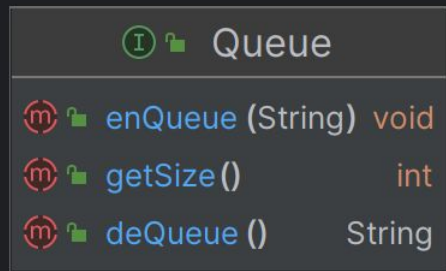


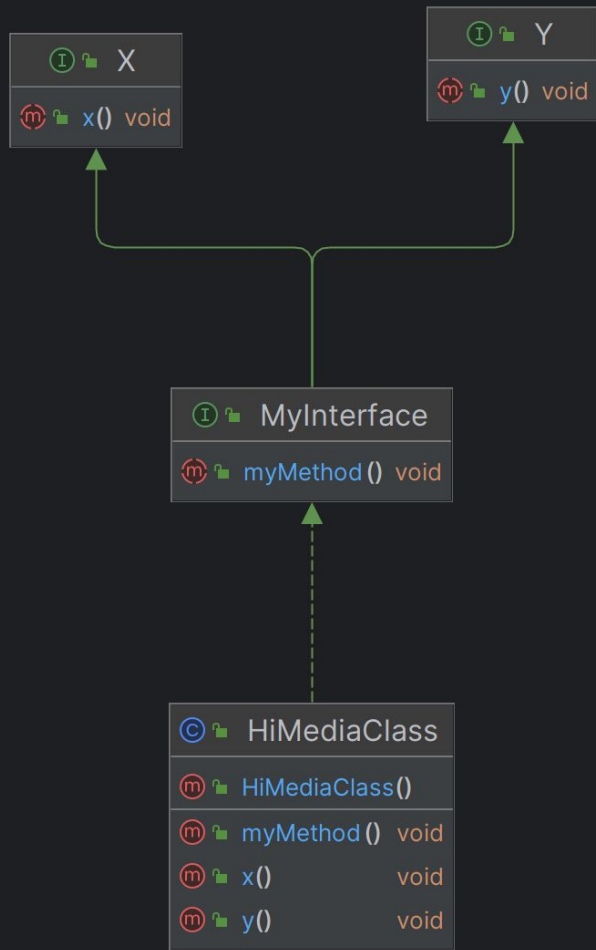
## Downcasting

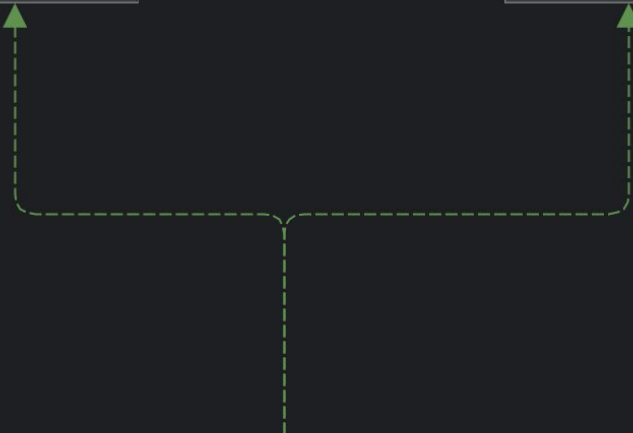
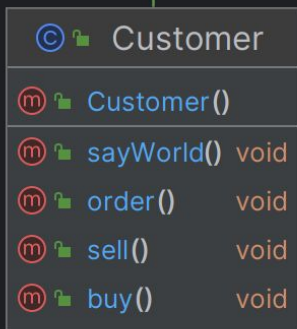
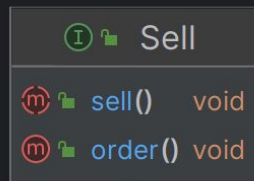
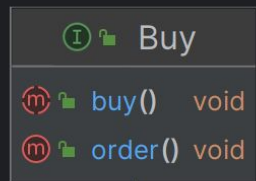




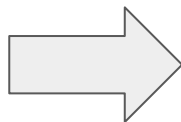
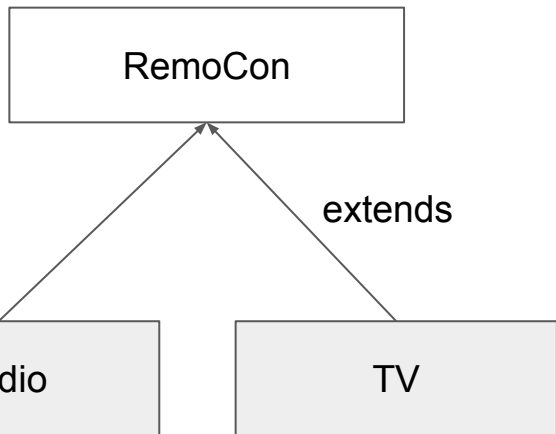




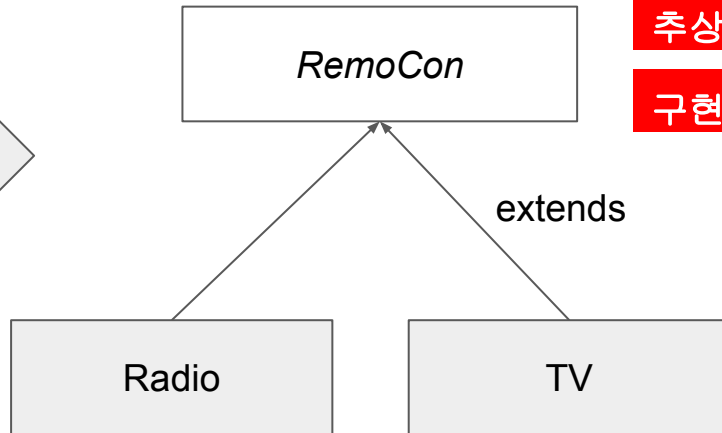




일반클래스



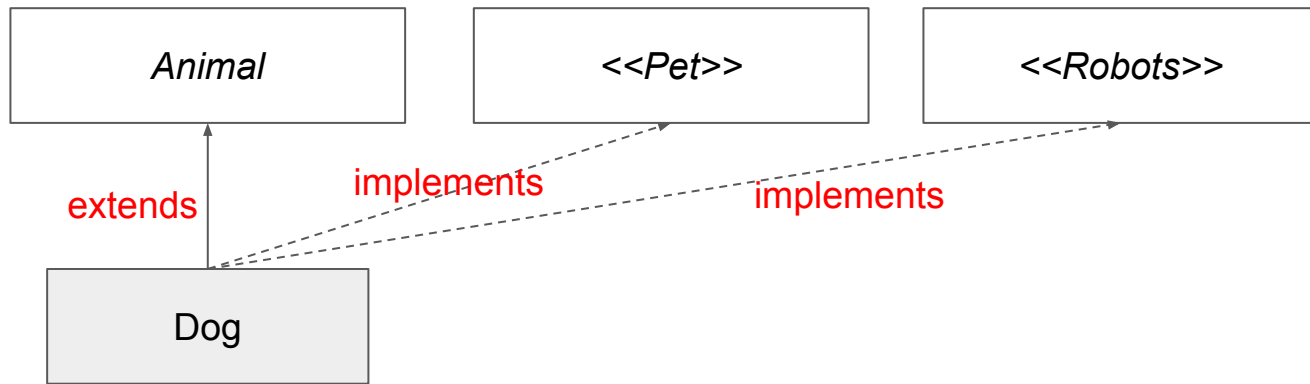
추상클래스



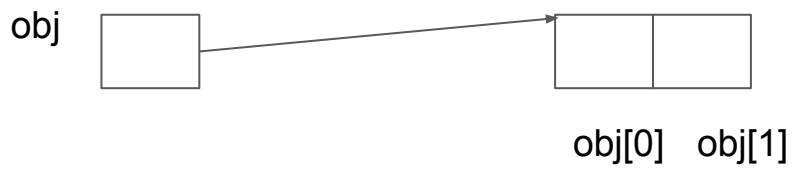
추상메서드

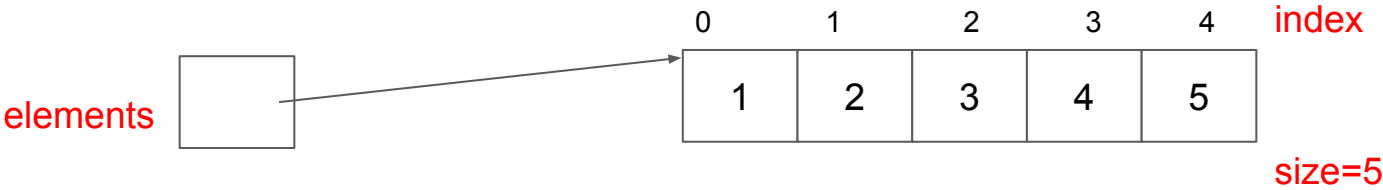
구현메서드

추상클래스

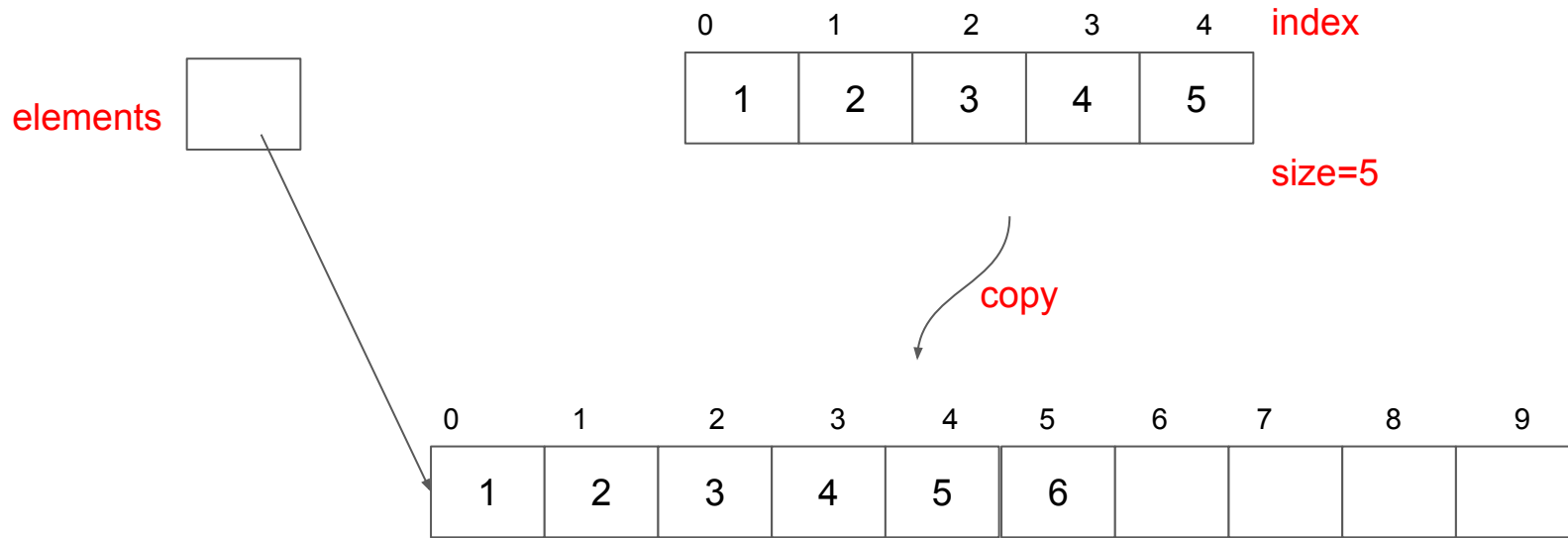


```
public class Dog extends Animal implements Pet, Robots {  
  
}
```









elements



자바	200 00	미디 어	이순 신
----	-----------	---------	---------

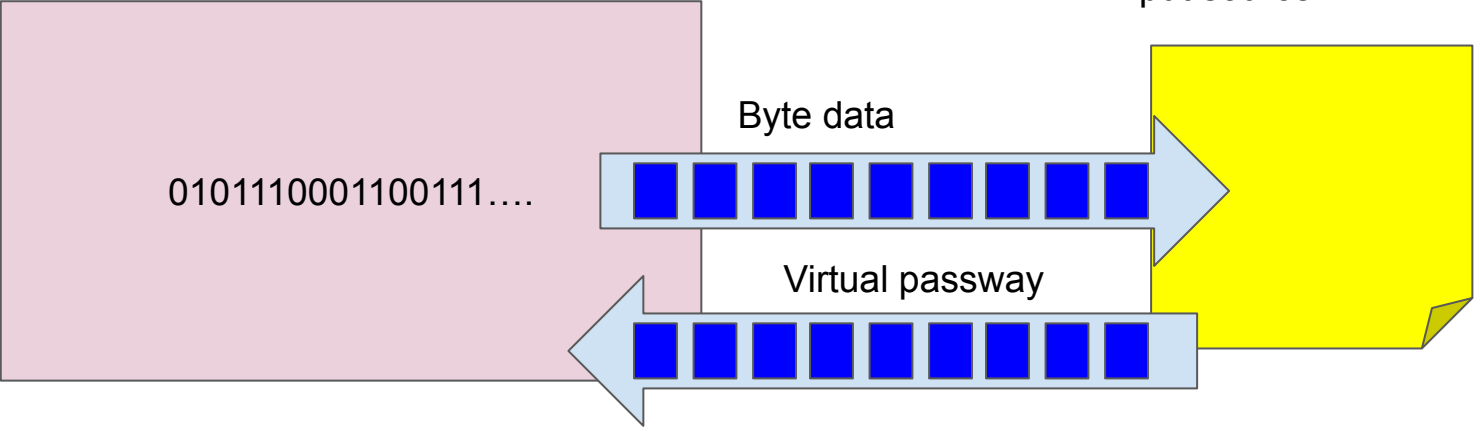
0 1 2 3 4 index

--	--	--	--	--

size=5

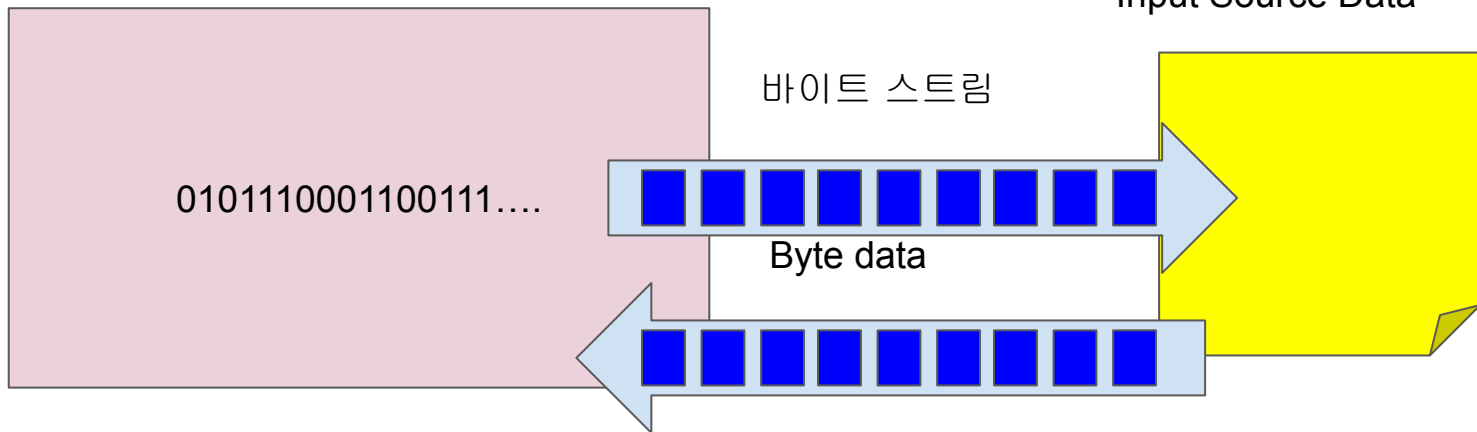
Java application

Output destination,  
Input source



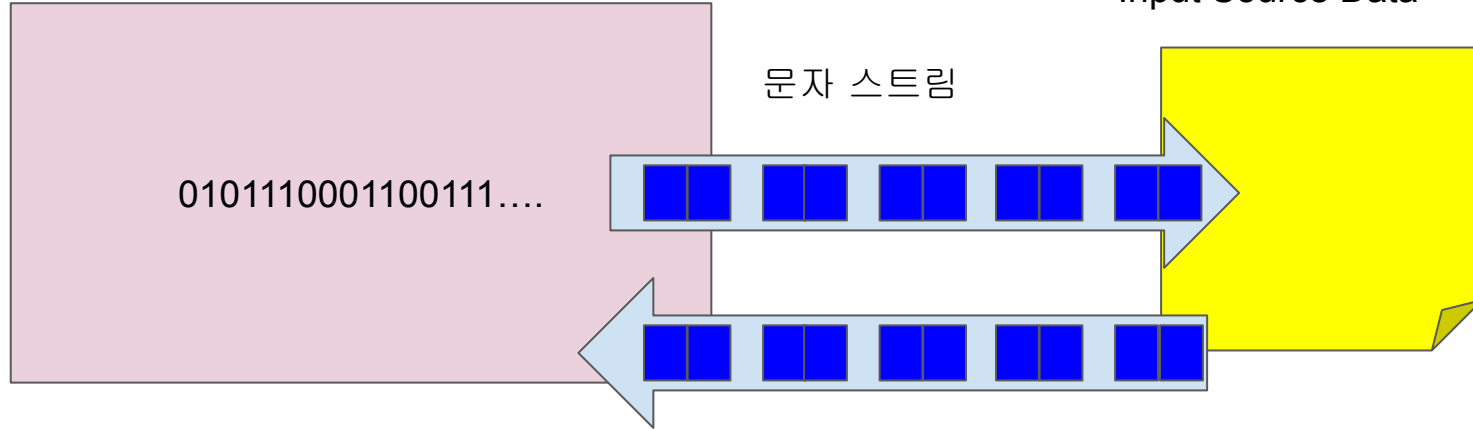
Java application

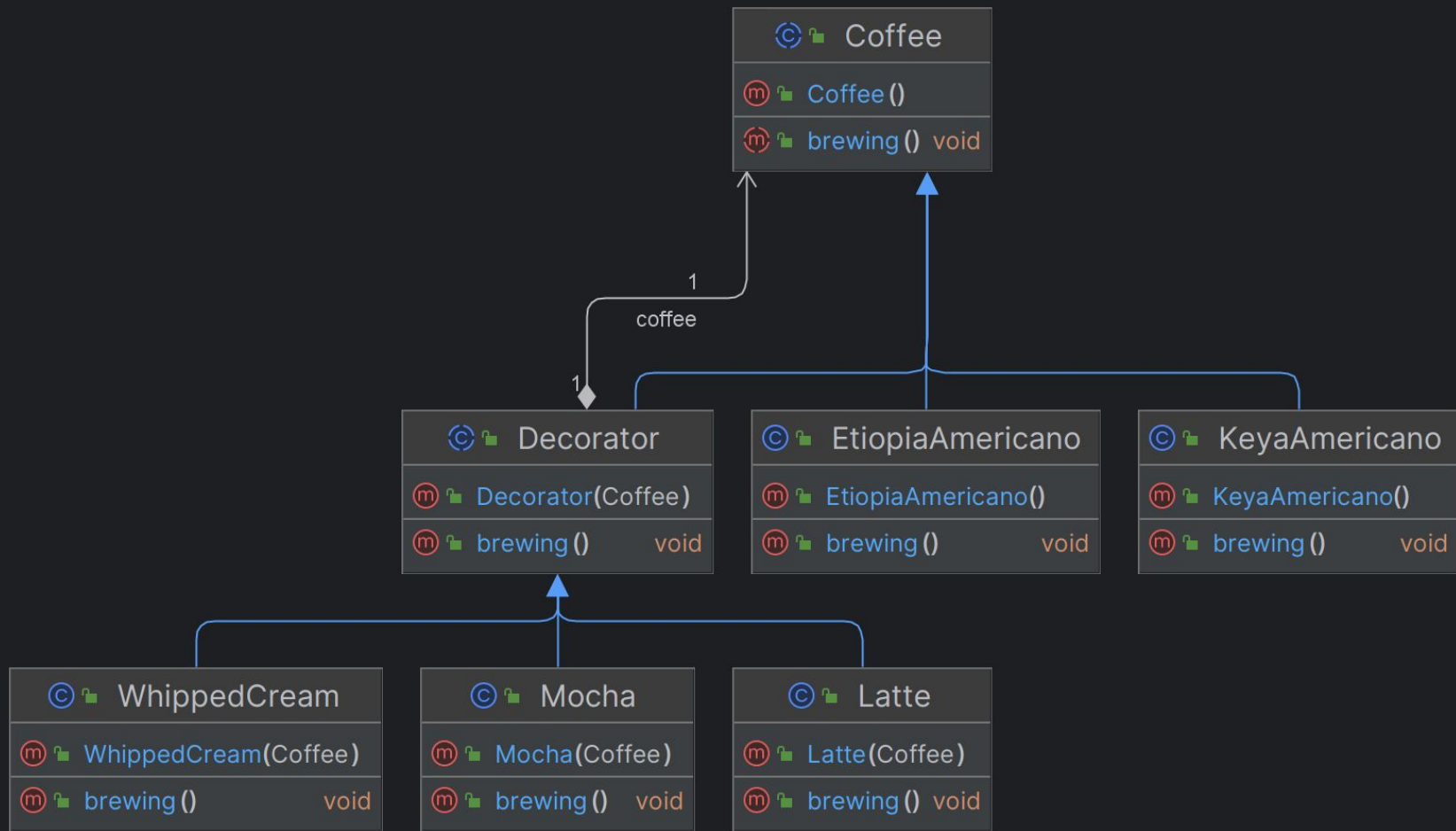
Output Destination Data,  
Input Source Data



Java application

Output Destination Data,  
Input Source Data





Thread Status

