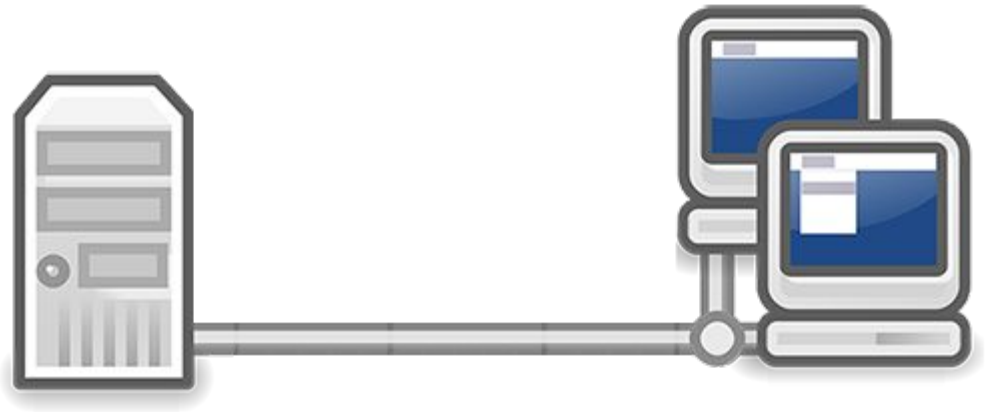
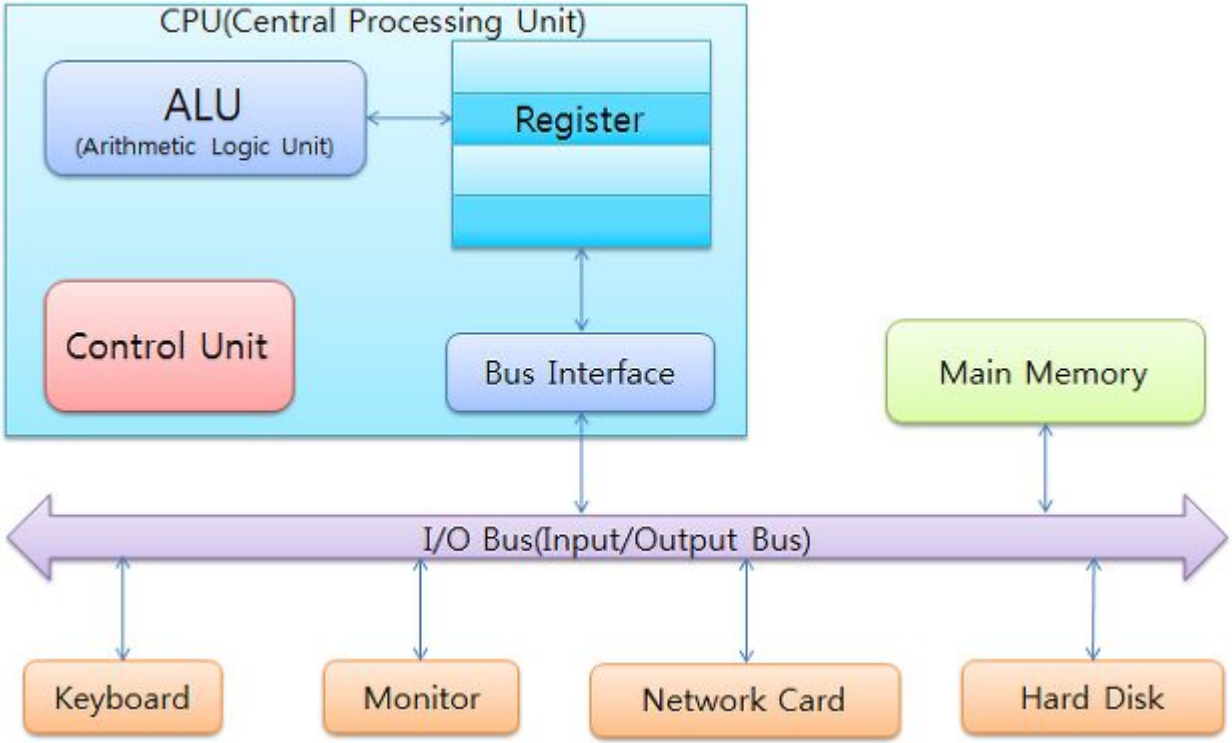
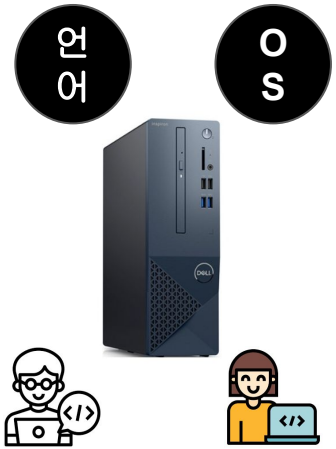
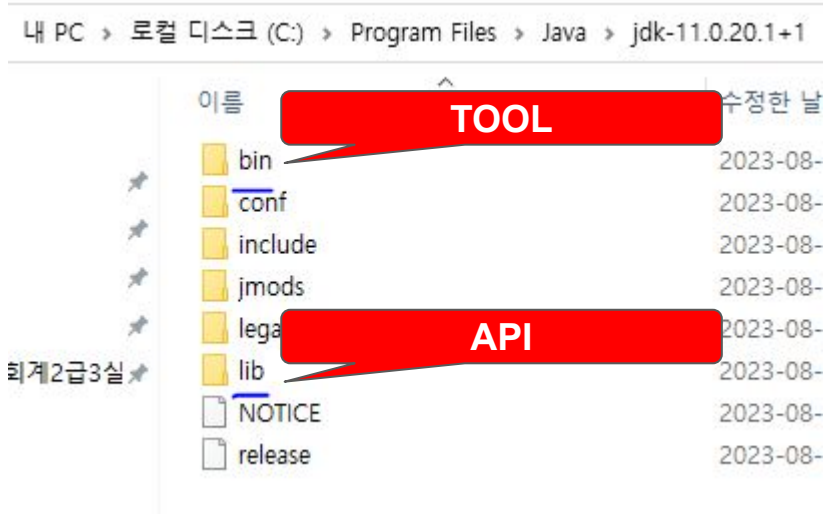
Java


서버
인터넷
클라이언트


CPU(Central Processing Unit)
ALU
(Arithmetic Logic Unit)
Register
Control Unit
Bus Interface
Main Memory
I/O Bus(Input/Output Bus)
Keyboard
Monitor
Network Card
Hard Disk


언어
OS

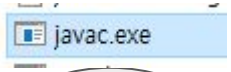

내 PC › 로컬 디스크 (C:) › Program Files › Java › jdk-11.0.20.1+1
이름
수정한 날
TOOL
bin
2023-08-
conf
2023-08-
include
2023-08-
jmods
2023-08-
lega
2023-08-
API
회계2급3실
lib
2023-08-
NOTICE
2023-08-
release
2023-08-


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

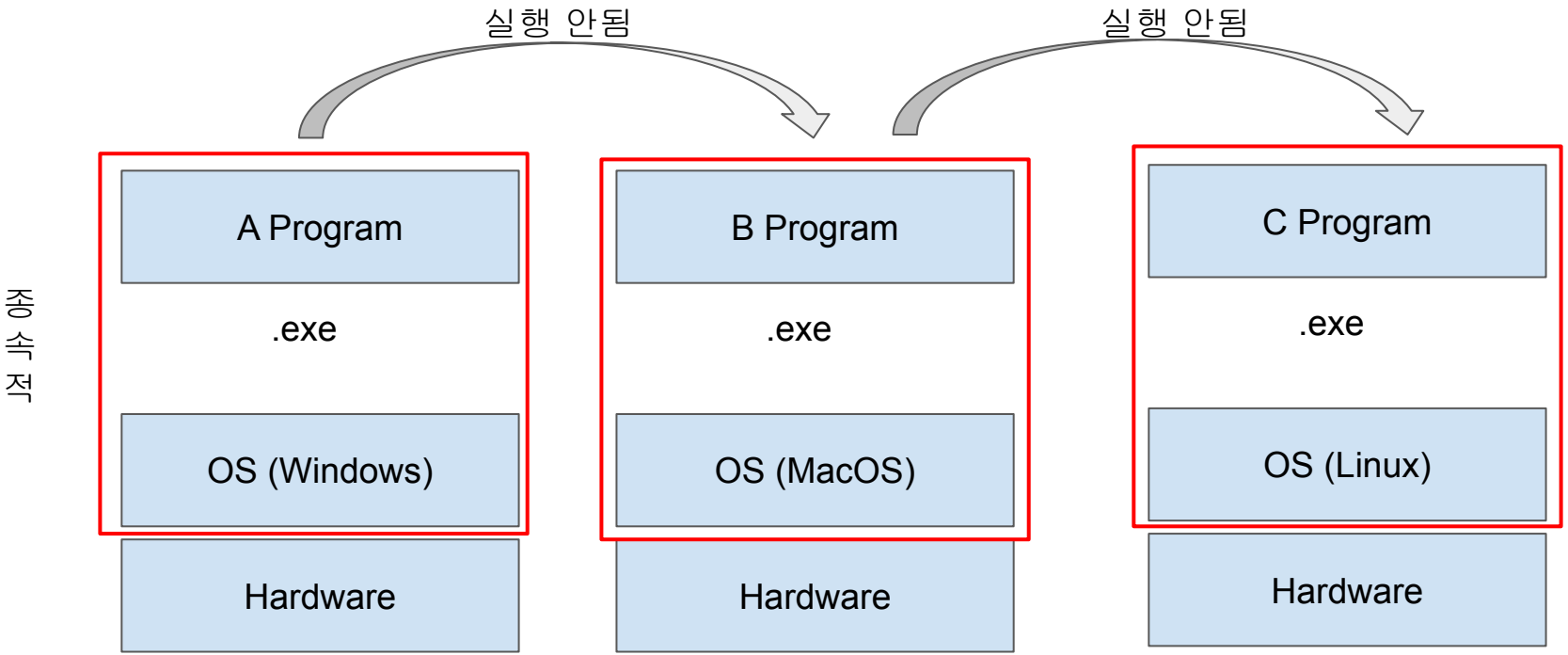
JDK에 포함된
컴파일러

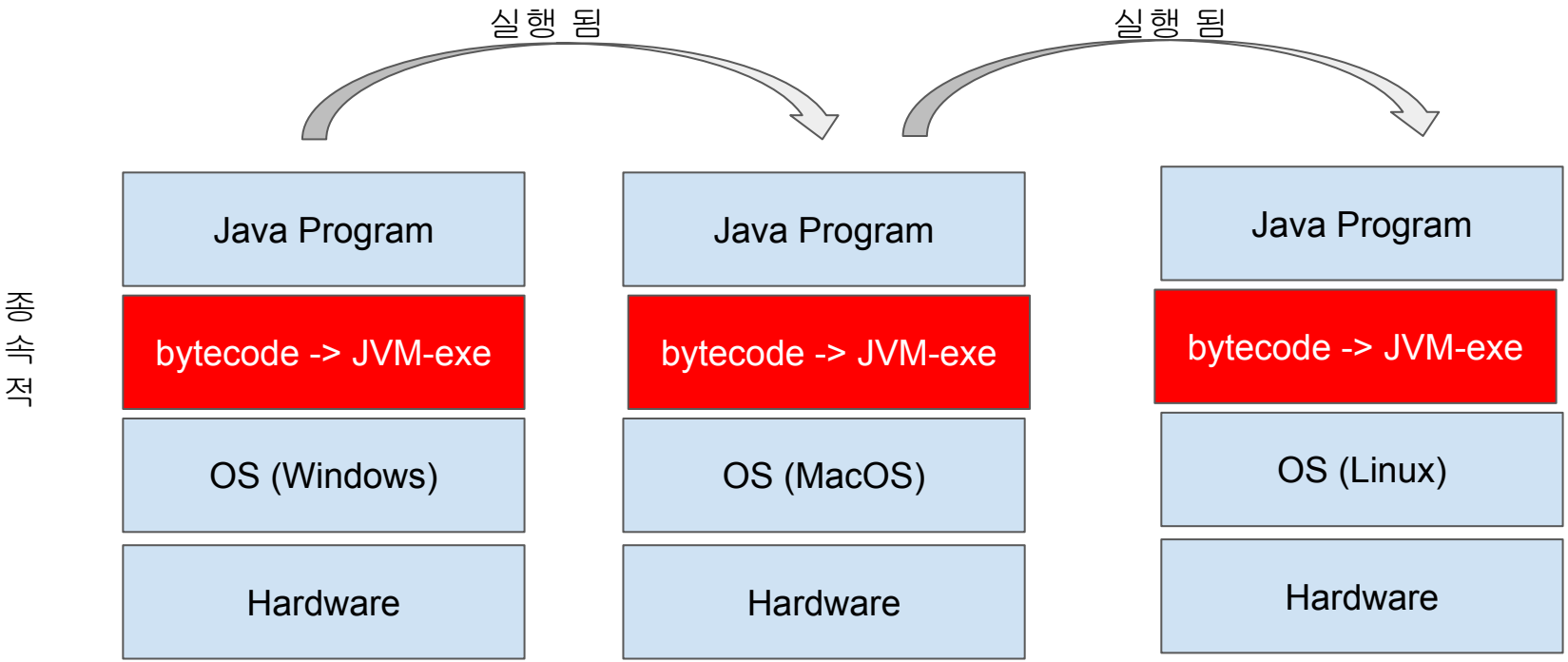


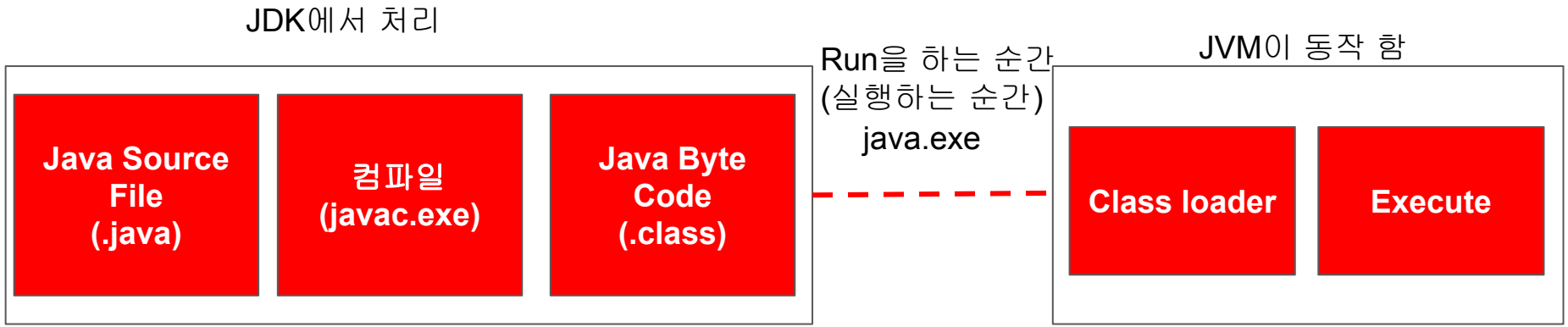
컴파일러

JDK에 포함된 실행

0010101111000011010
0001101000000001111
0000111100111100111

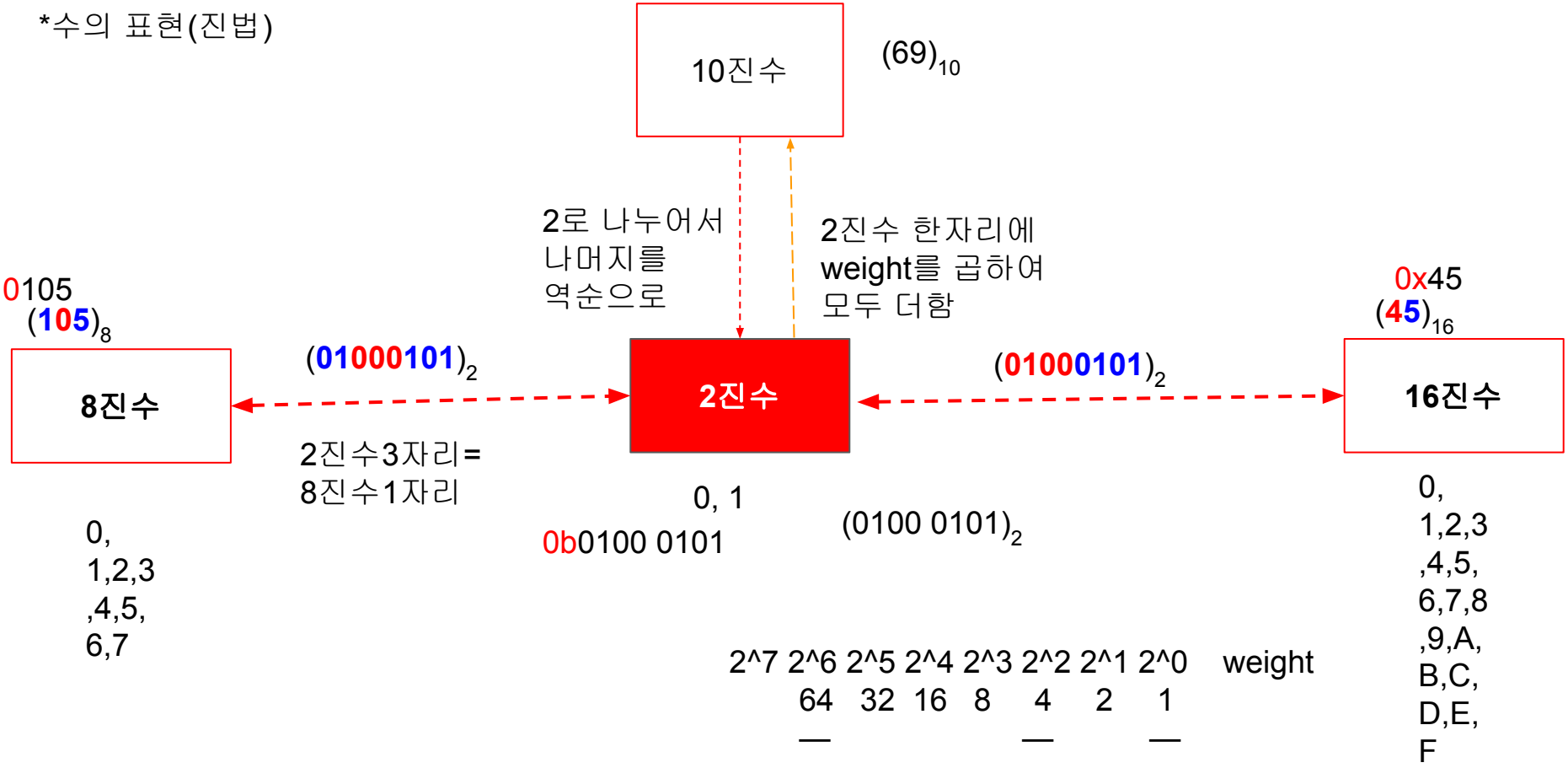

실행 안됨
실행 안됨
A Program
.exe
OS (Windows)
Hardware
B Program
.exe
OS (MacOS)
Hardware
C Program
.exe
OS (Linux)
Hardware


실행 됨
실행 됨
Java Program
bytecode -> JVM-exe
OS (Windows)
Hardware
Java Program
bytecode -> JVM-exe
OS (MacOS)
Hardware
Java Program
bytecode -> JVM-exe
OS (Linux)
Hardware
좋은 속도 적

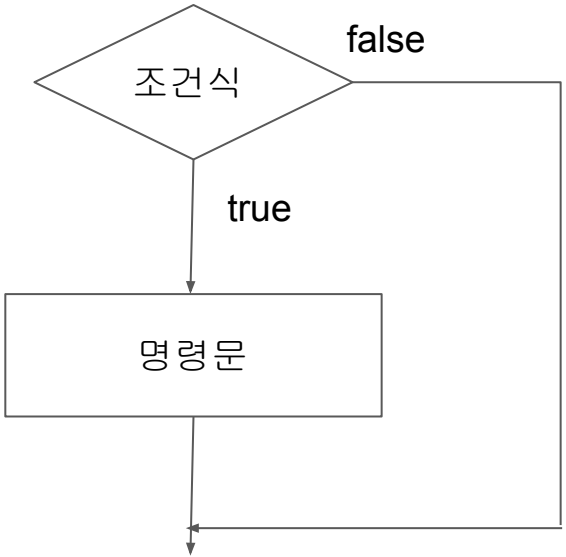


java.exe Hello (실행)
⇒ 원격프로세서 호출 (JVM)

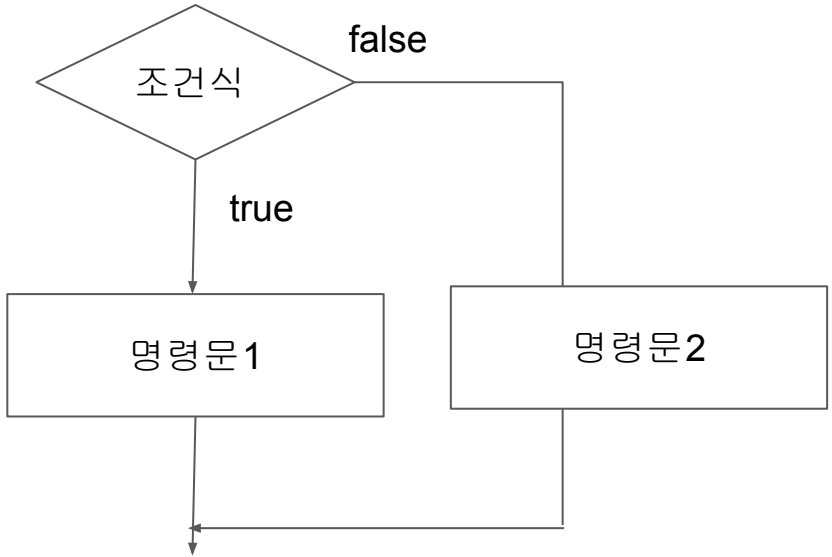
*수의 표현(진법)

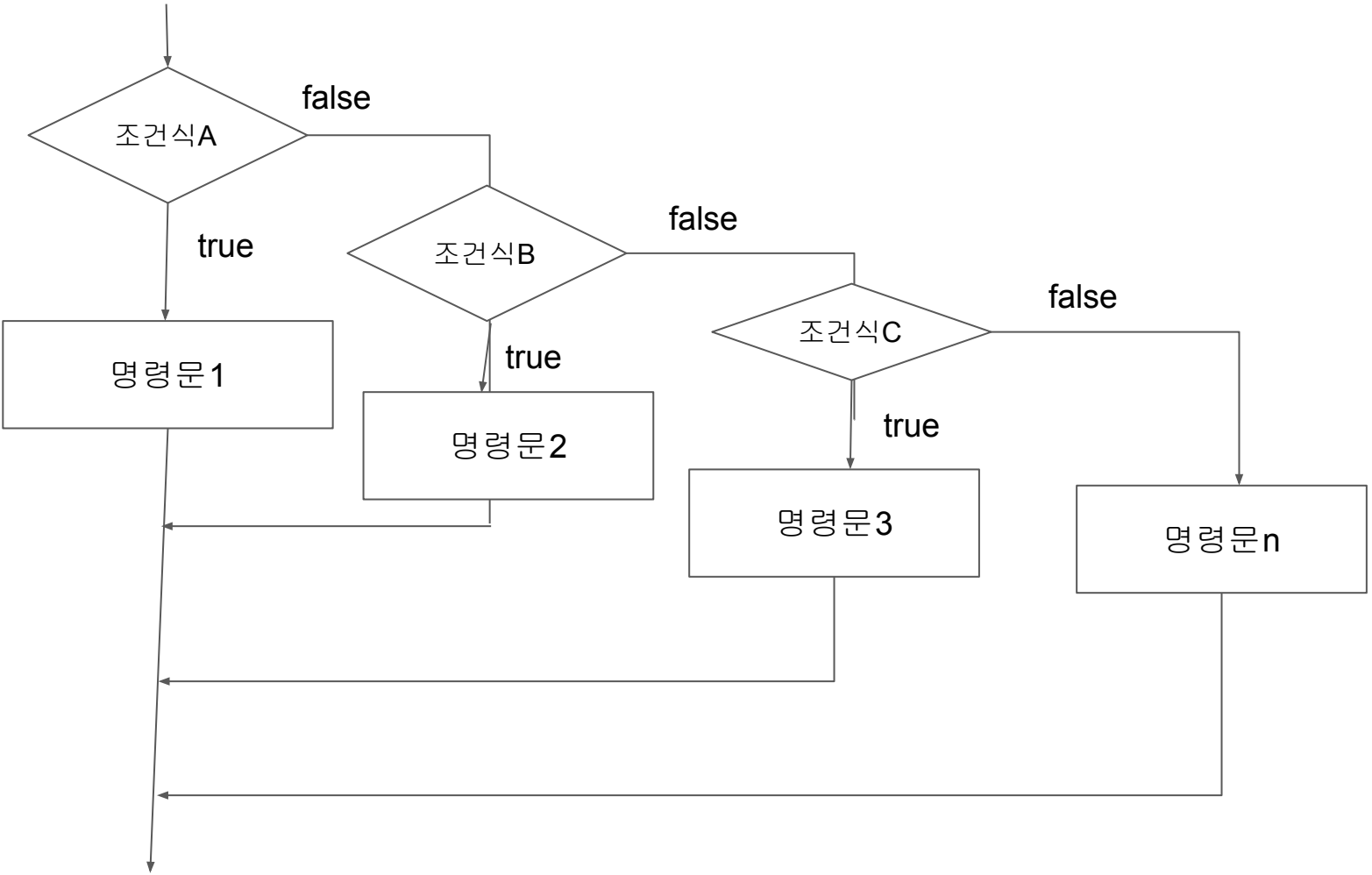

10진수
$(69)_{10}$
2로 나누어서
나머지를
역순으로
2진수 한자리에
weight를 곱하여
모두 더함
0105
$(105)_8$
8진수
$(01000101)_2$
2진수3자리=
8진수1자리
0,
1,2,3
,4,5,
6,7
2진수
0, 1
0b0100 0101
$(01000101)_2$
$(0100\ 0101)_2$
0x45
$(45)_{16}$
16진수
0,
1,2,3
,4,5,
6,7,8
,9,A,
B,C,
D,E,
F
2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0 weight
64 32 16 8 4 2 1

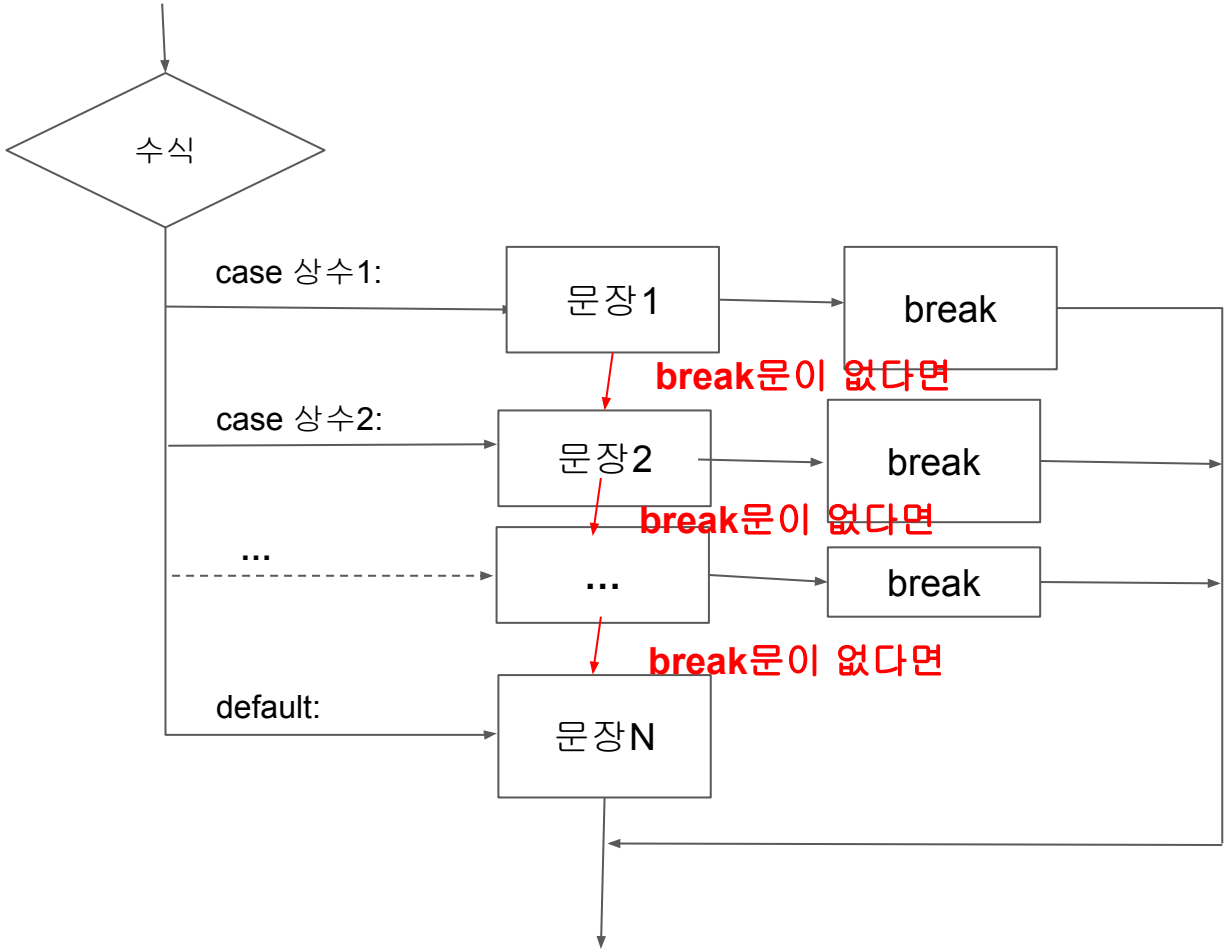
if문



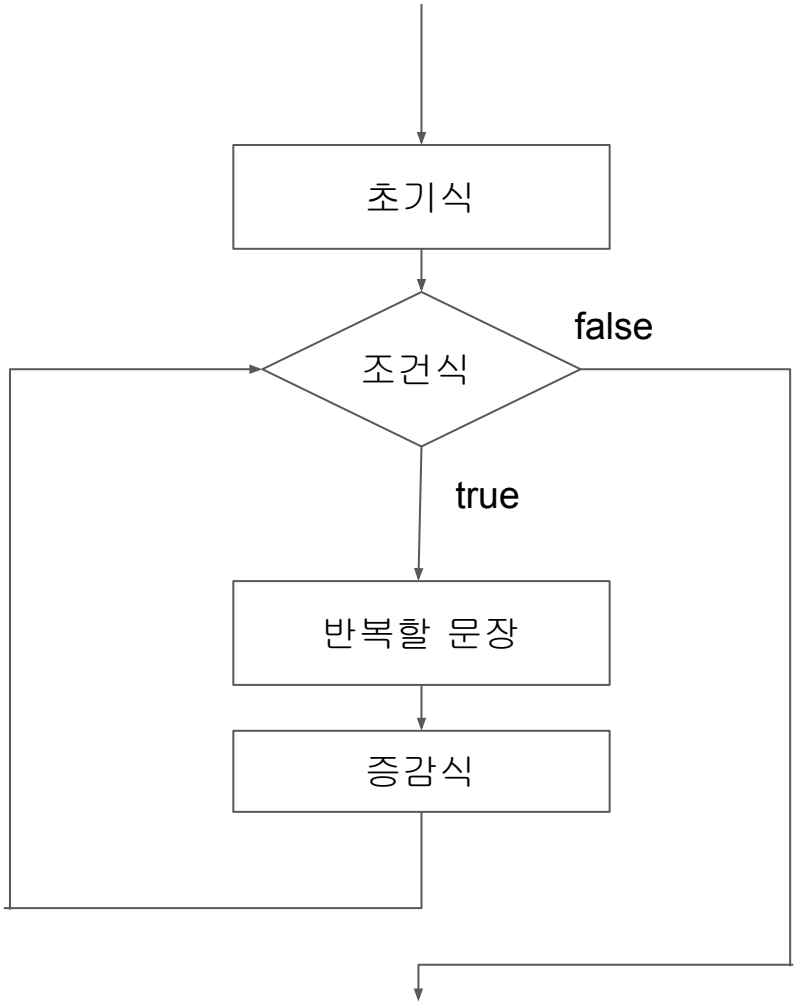
if ~ else 문



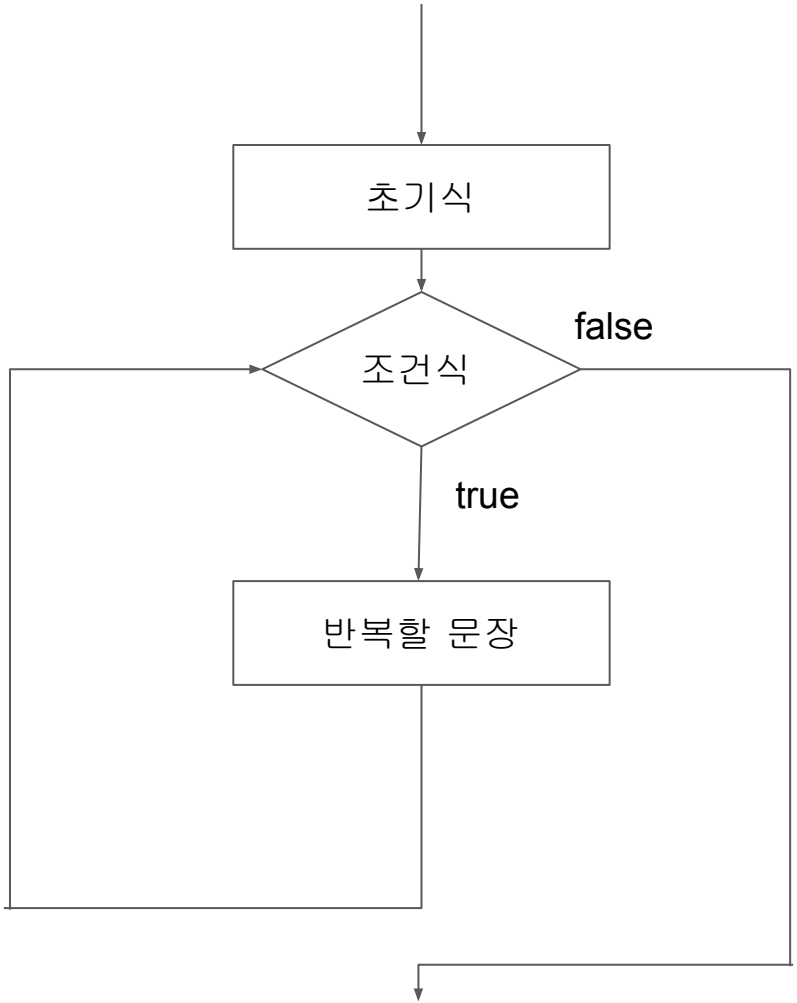

조건식A
false
true
조건식B
false
true
조건식C
false
true
명령문1
명령문2
명령문3
명령문n


수식
case 상수1:
문장1
break
break문이 없다면
case 상수2:
문장2
break
break문이 없다면
...
...
break
break문이 없다면
default:
문장N

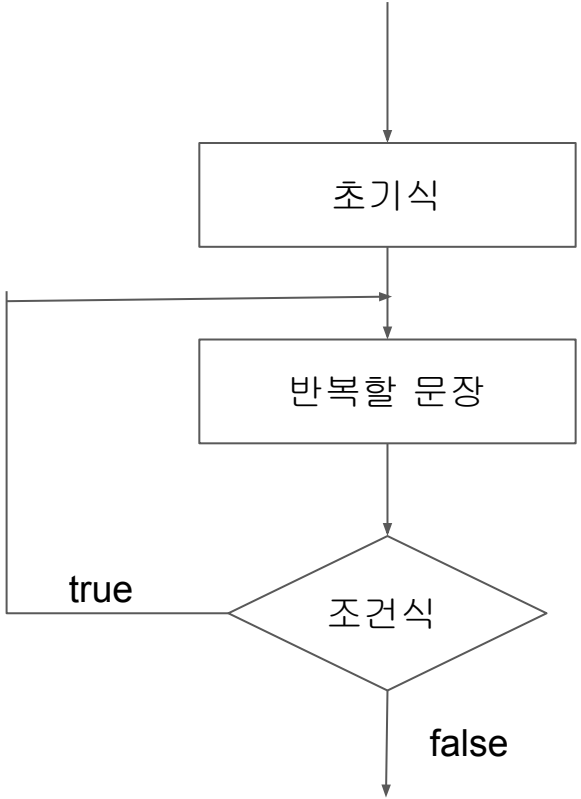
for문

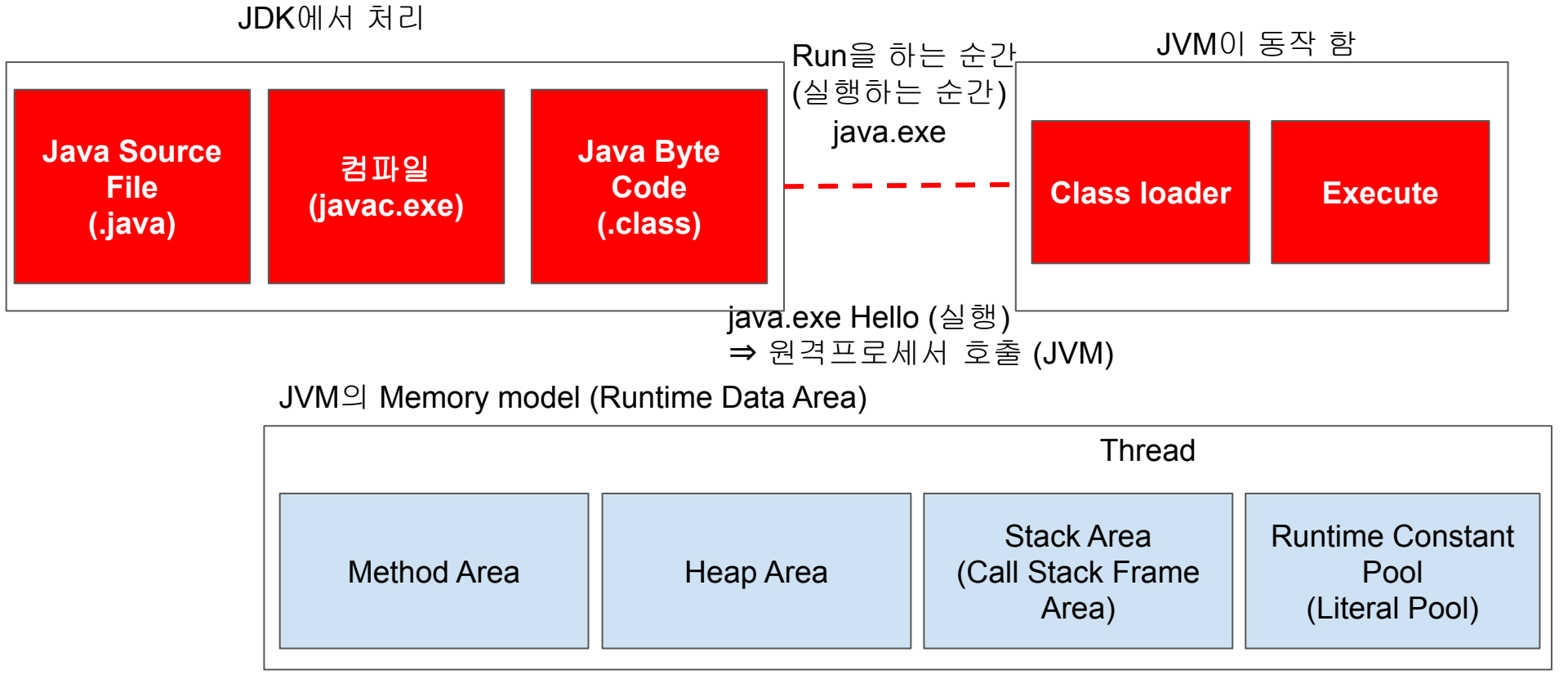


while문

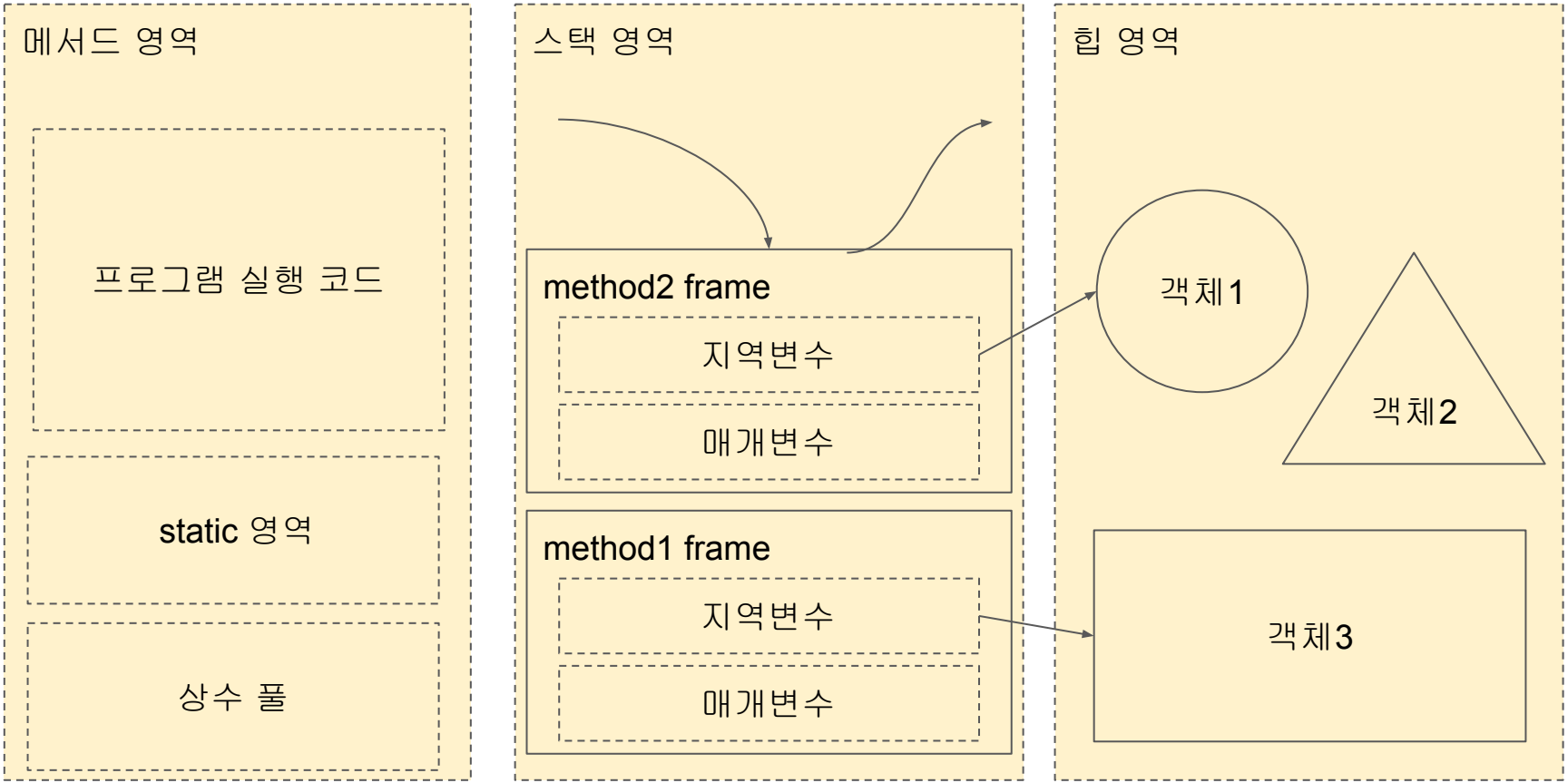


do-while문

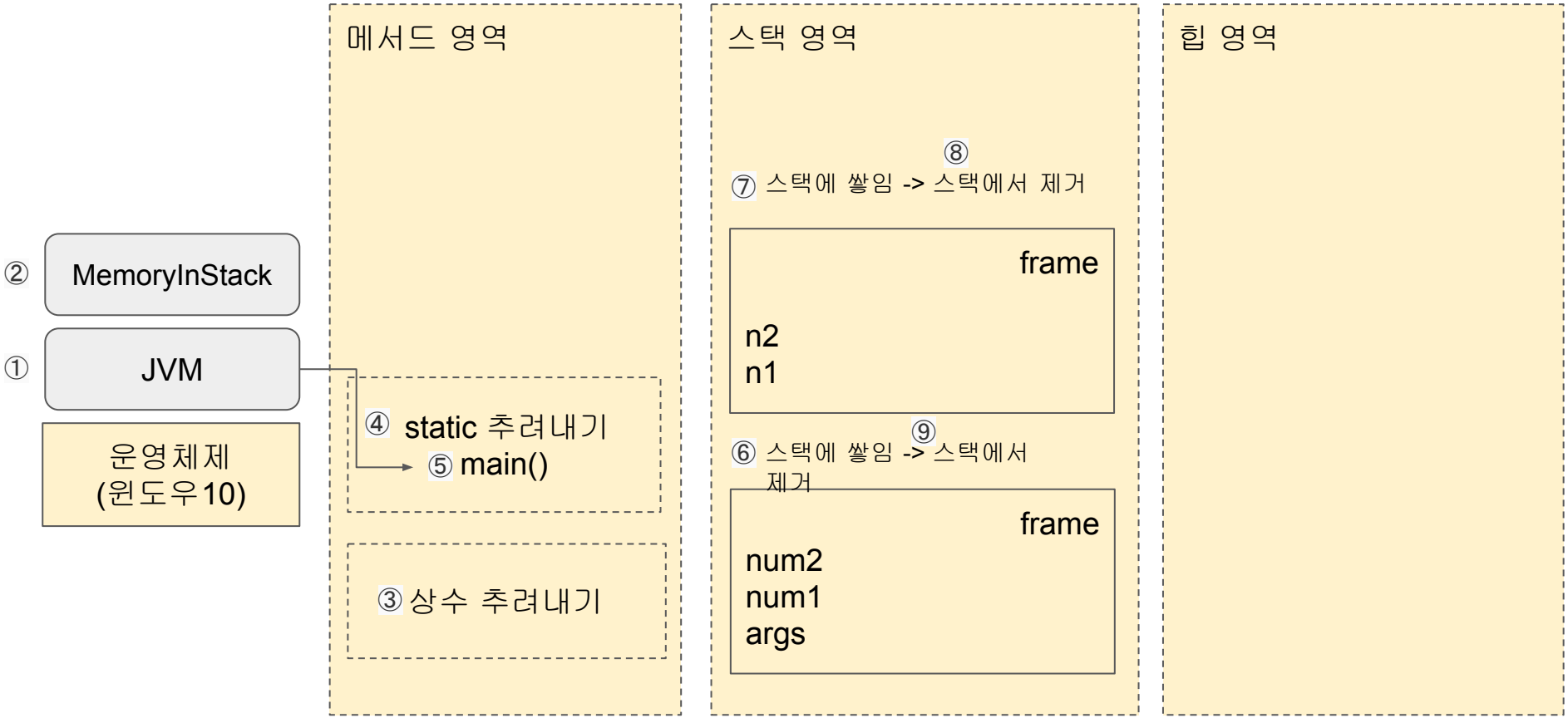


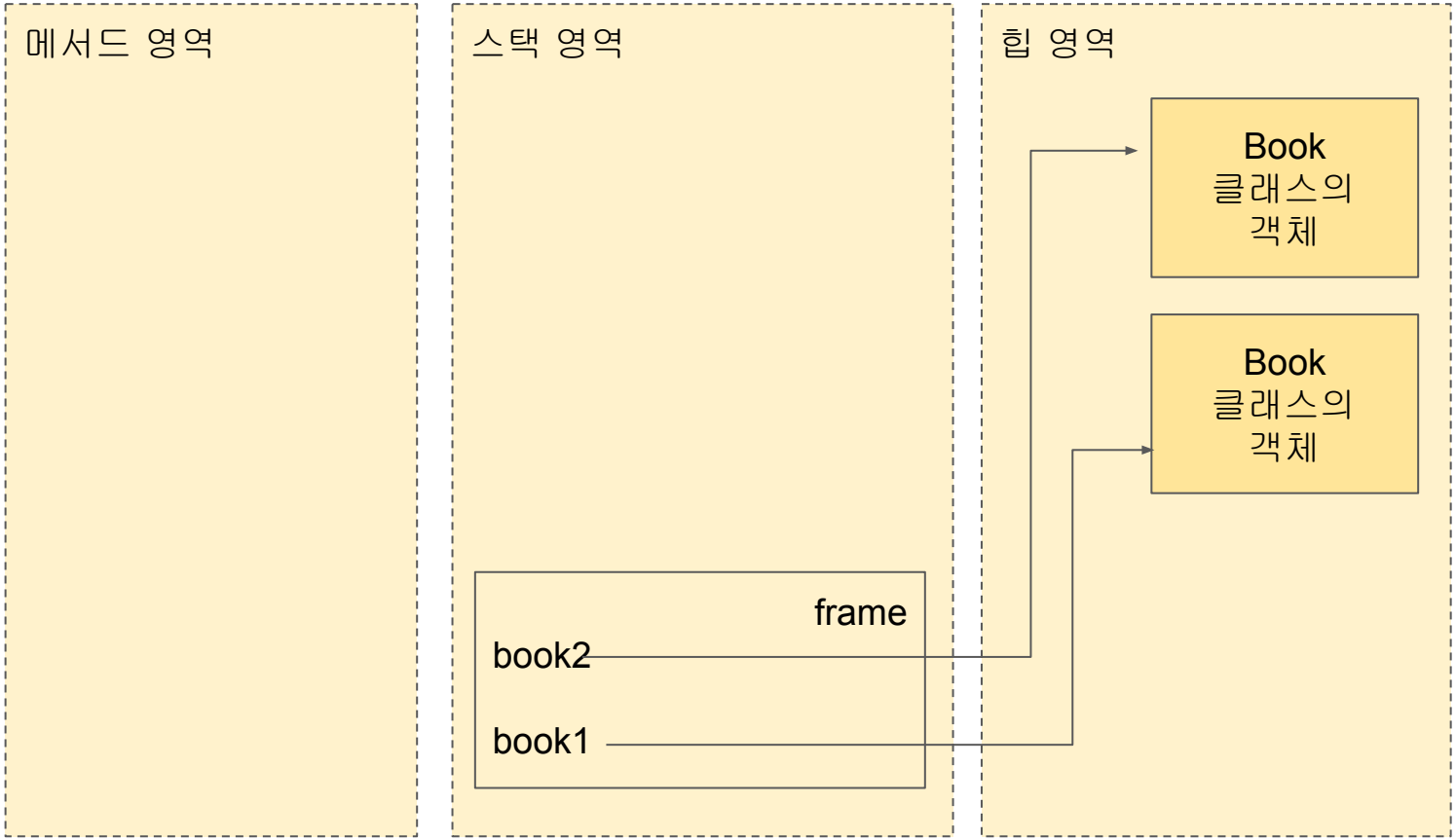

JDK에서 처리
Java Source File (.java)
컴파일 (javac.exe)
Java Byte Code (.class)
Run을 하는 순간
(실행하는 순간)
java.exe
JVM이 동작 함
Class loader
Execute
java.exe Hello (실행)
⇒ 원격프로세서 호출 (JVM)
JVM의 Memory model (Runtime Data Area)
Thread
Method Area
Heap Area
Stack Area (Call Stack Frame Area)
Runtime Constant Pool (Literal Pool)

런타임 data 영역들

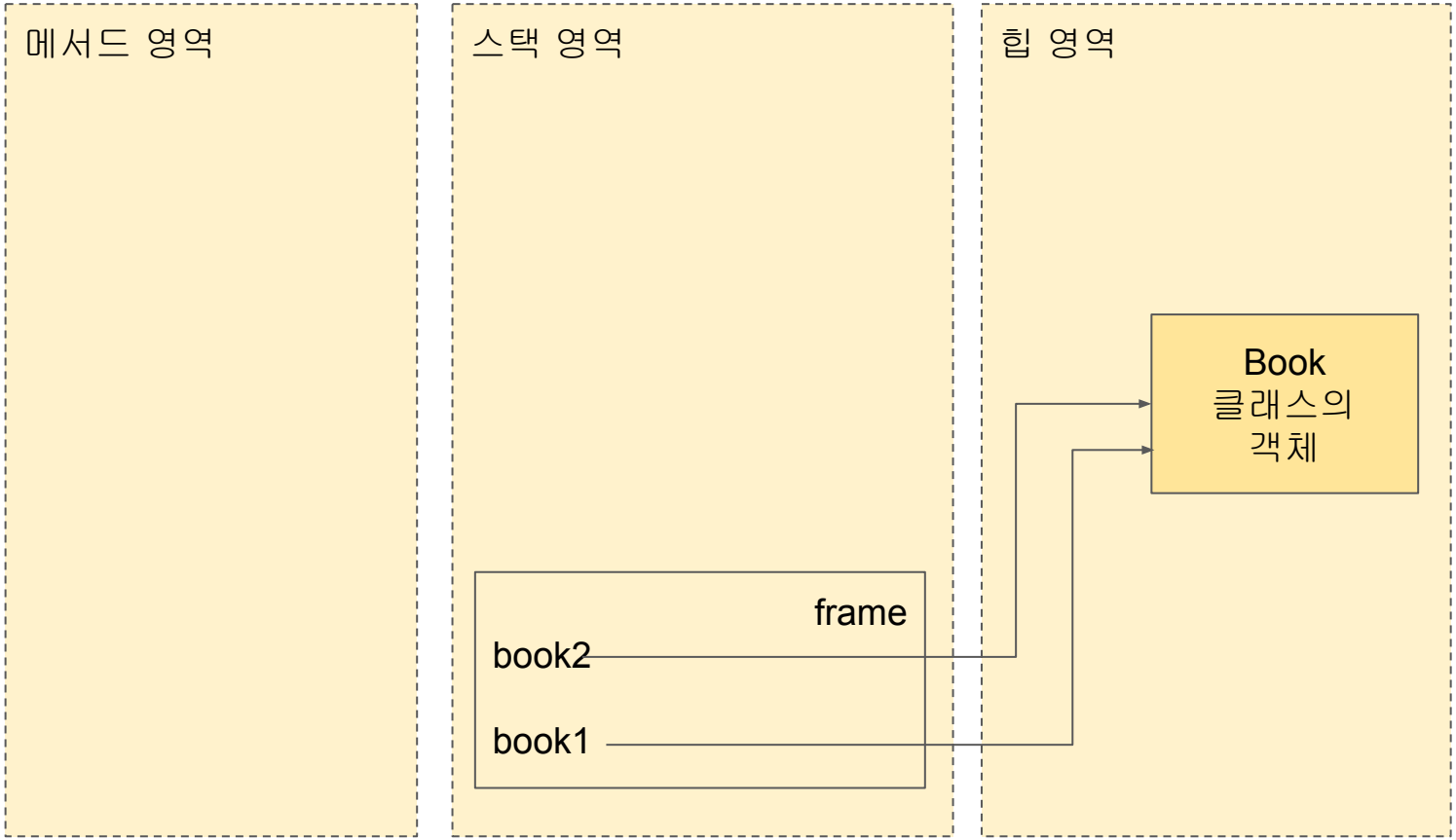

메서드 영역
프로그램 실행 코드
static 영역
상수 풀
스택 영역
method2 frame
지역변수
매개변수
method1 frame
지역변수
매개변수
힙 영역
객체1
객체2
객체3

①②③④⑤⑥⑦⑧⑨⑩⑪⑫⑬⑭⑮⑯⑰



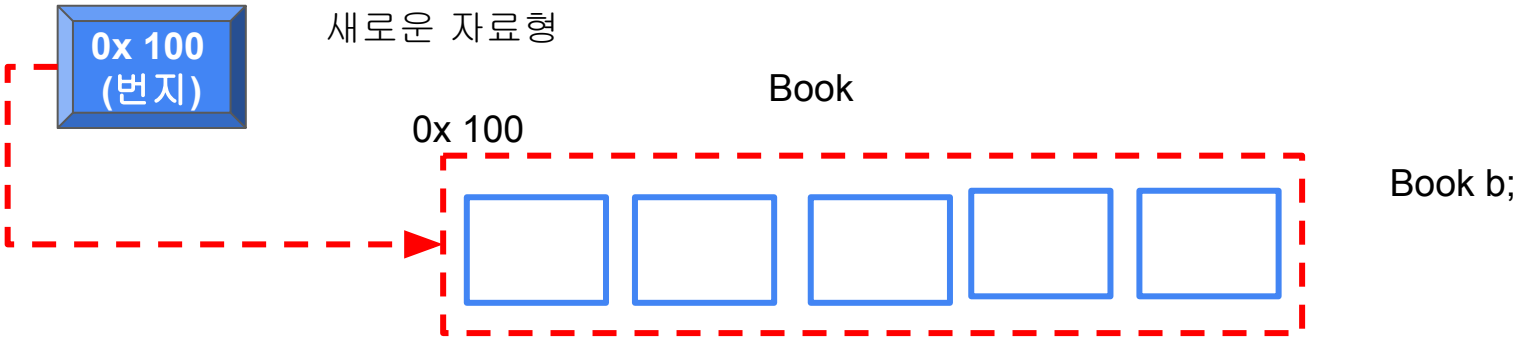

메서드 영역
스택 영역
힙 영역
Book
클래스의
객체
Book
클래스의
객체
frame
book2
book1

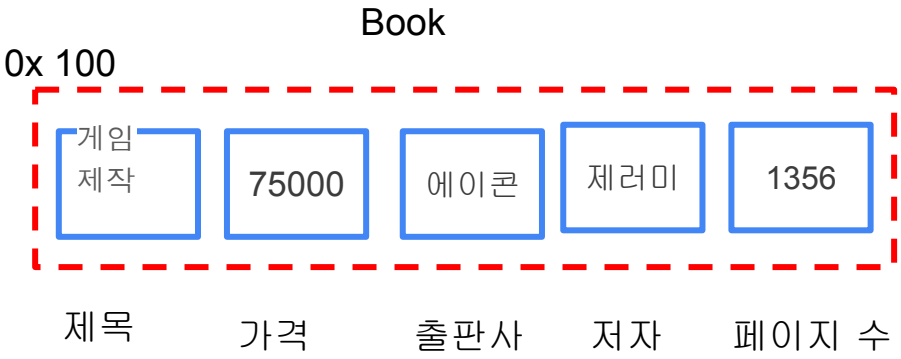
①②③④⑤⑥⑦⑧⑨⑩⑪⑫⑬⑭⑮⑯⑰



기본자료형

int a;

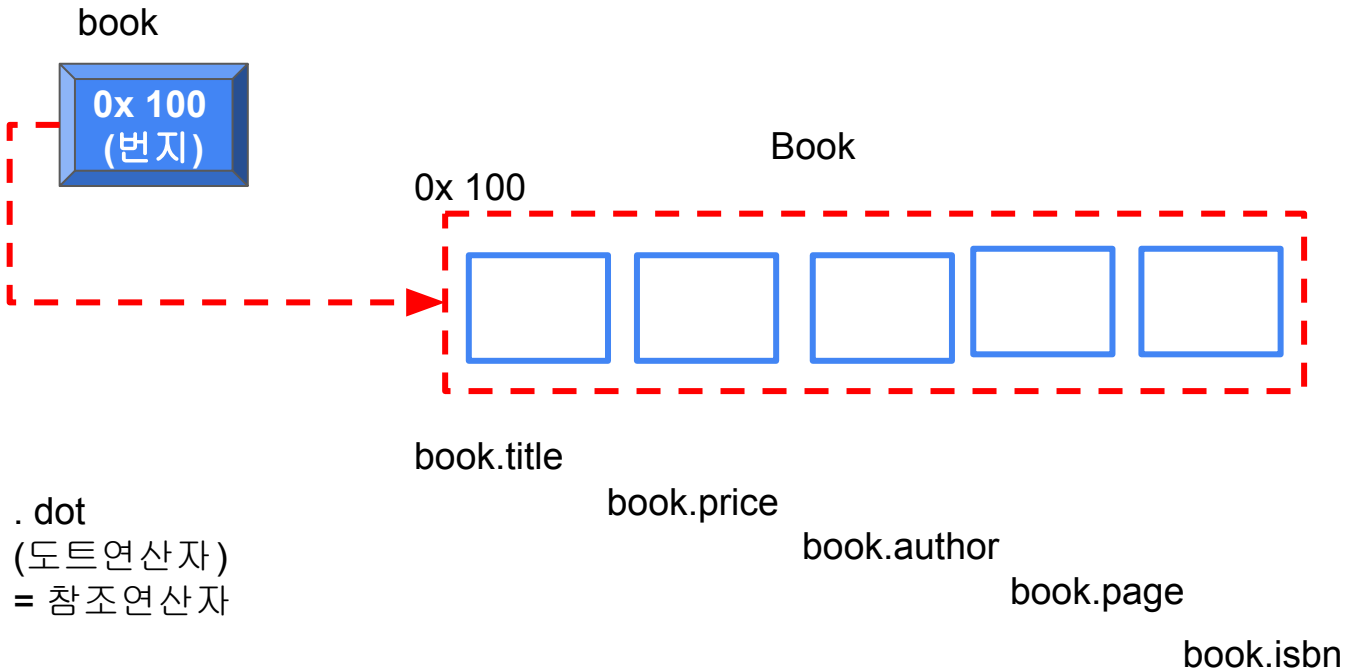




Book b;

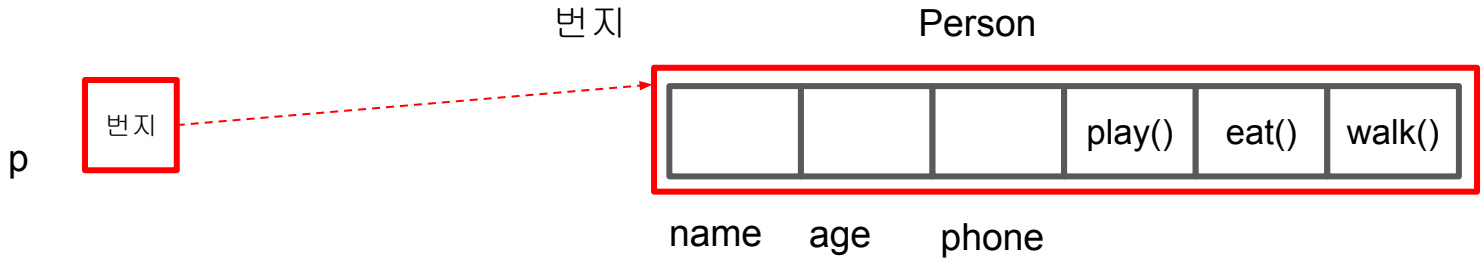
기억공간을 하나의 구조(객체)

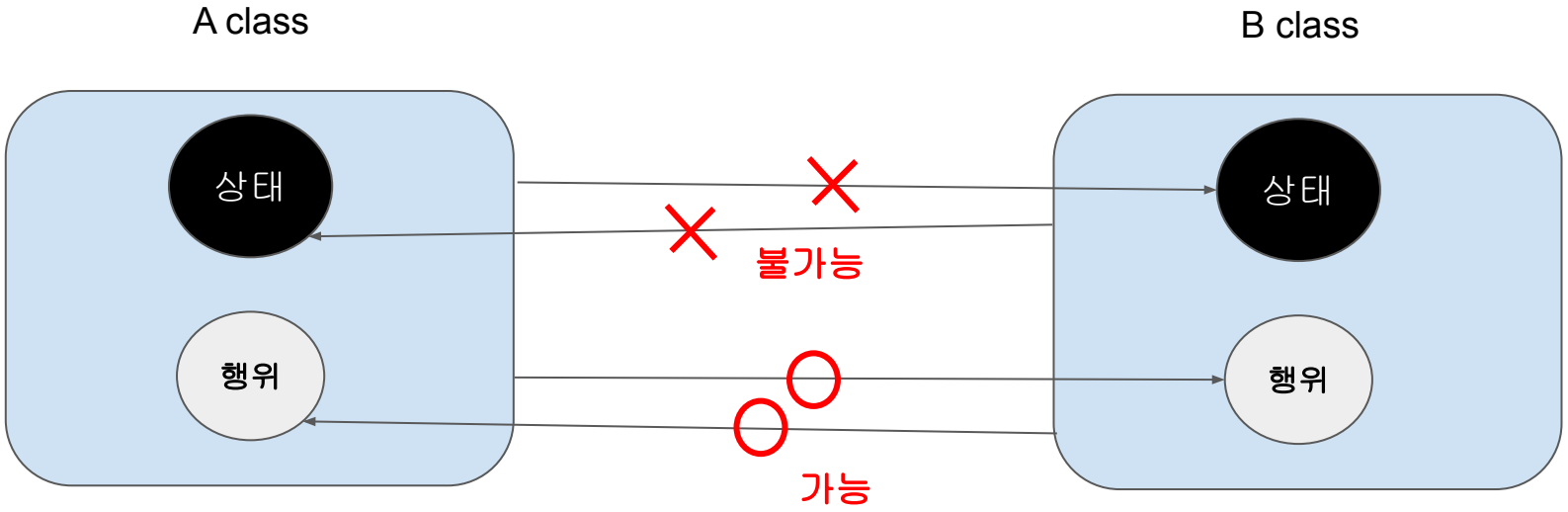
Book book = new Book();



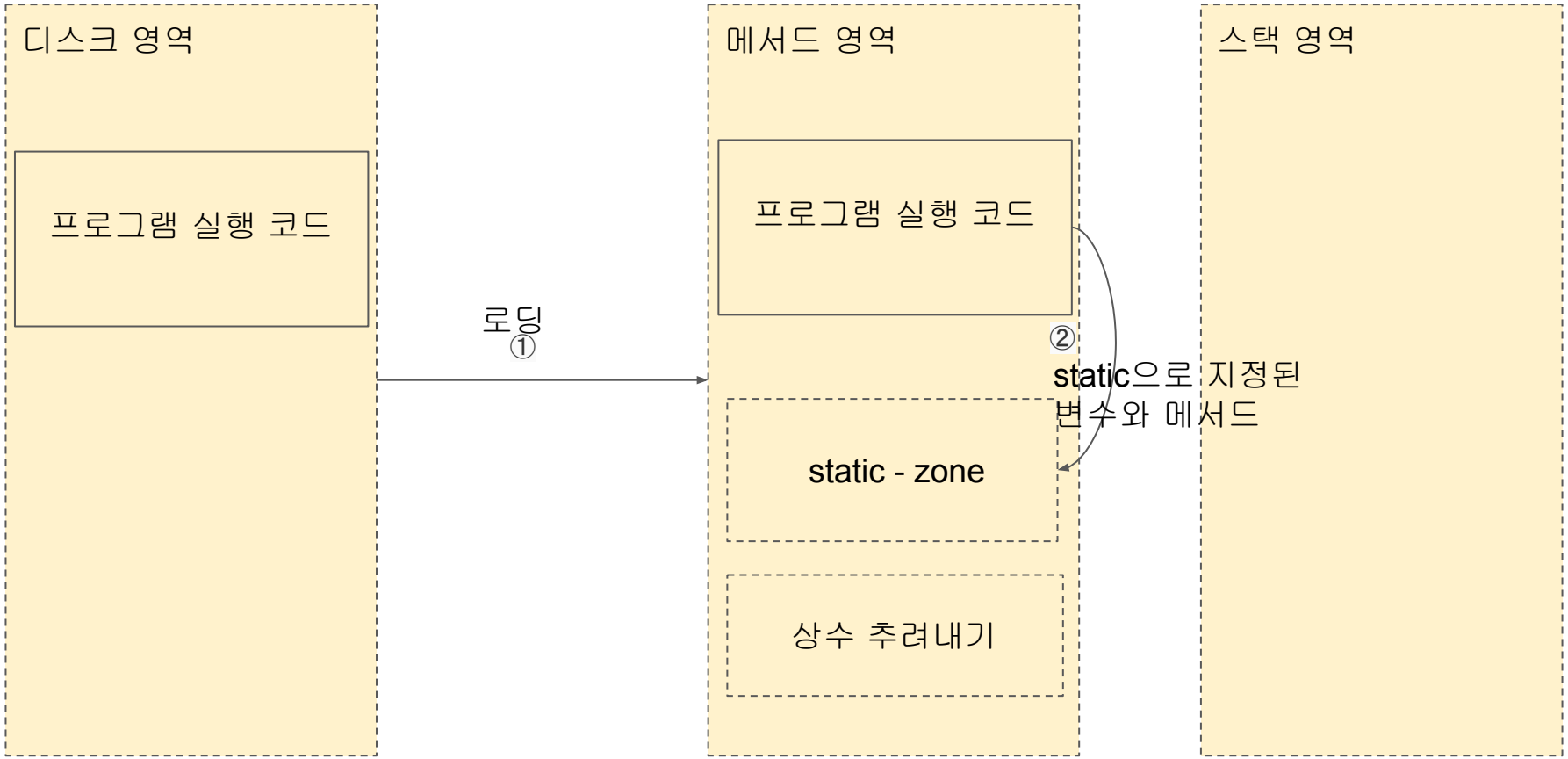
. dot
(도트연산자)
= 참조연산자

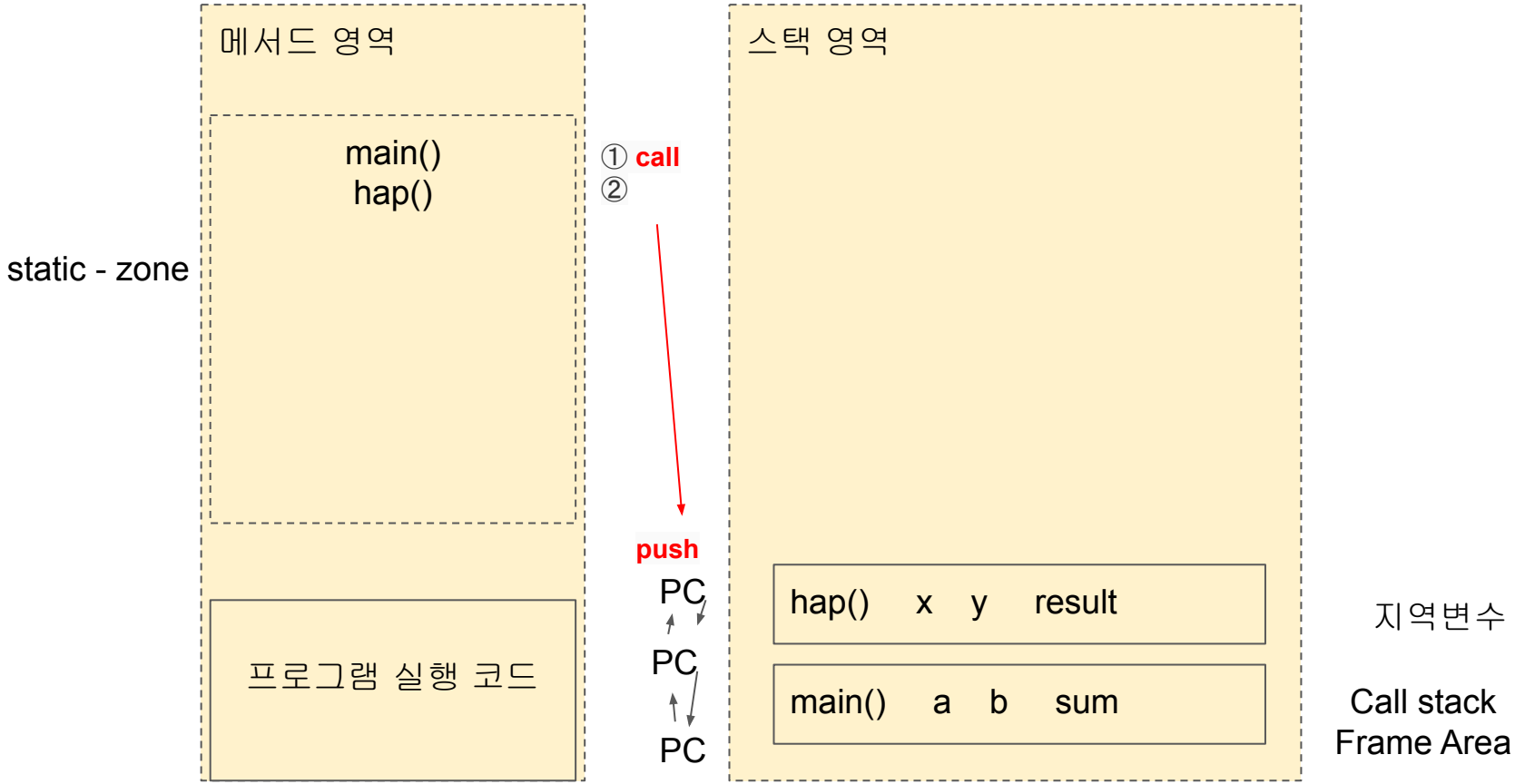
Person p = new Person();

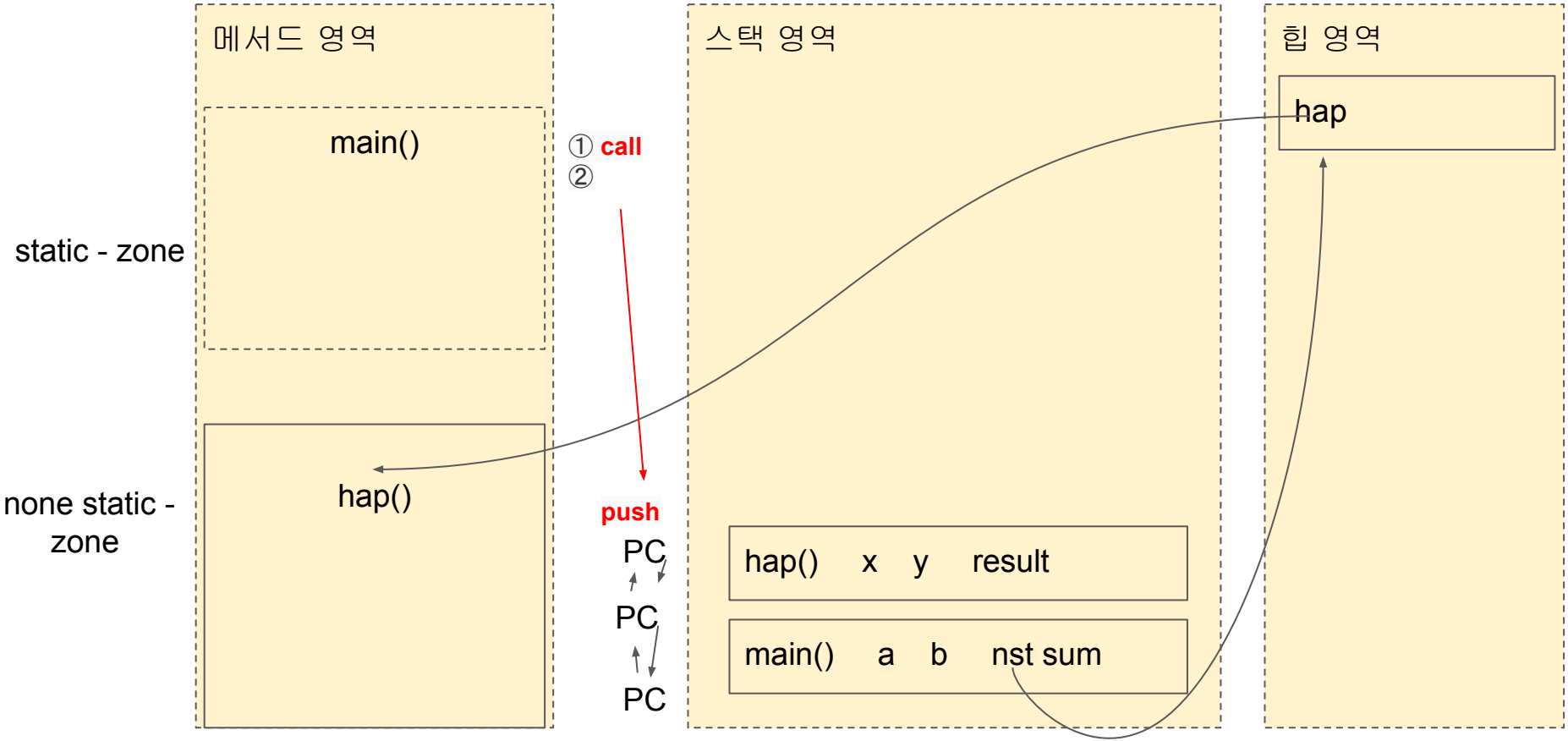



A class
B class
상태
행위
상태
행위
불가능
가능

①②③④⑤⑥⑦⑧⑨⑩⑪⑫⑬⑭⑮⑯⑰




메서드 영역
main()
hap()
static - zone
프로그램 실행 코드
① call
②
push
PC
PC
PC
스택 영역
hap() x y result
main() a b sum
지역변수
Call stack
Frame Area

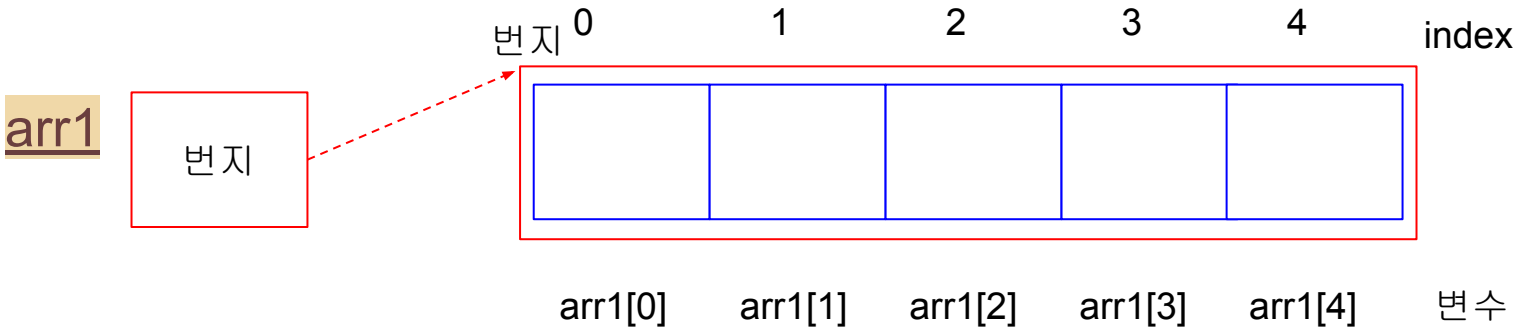

메서드 영역
스택 영역
힙 영역
main()
① call
②
hap
static - zone
hap()
none static - zone
push
PC
PC
PC
hap() x y result
main() a b nst sum

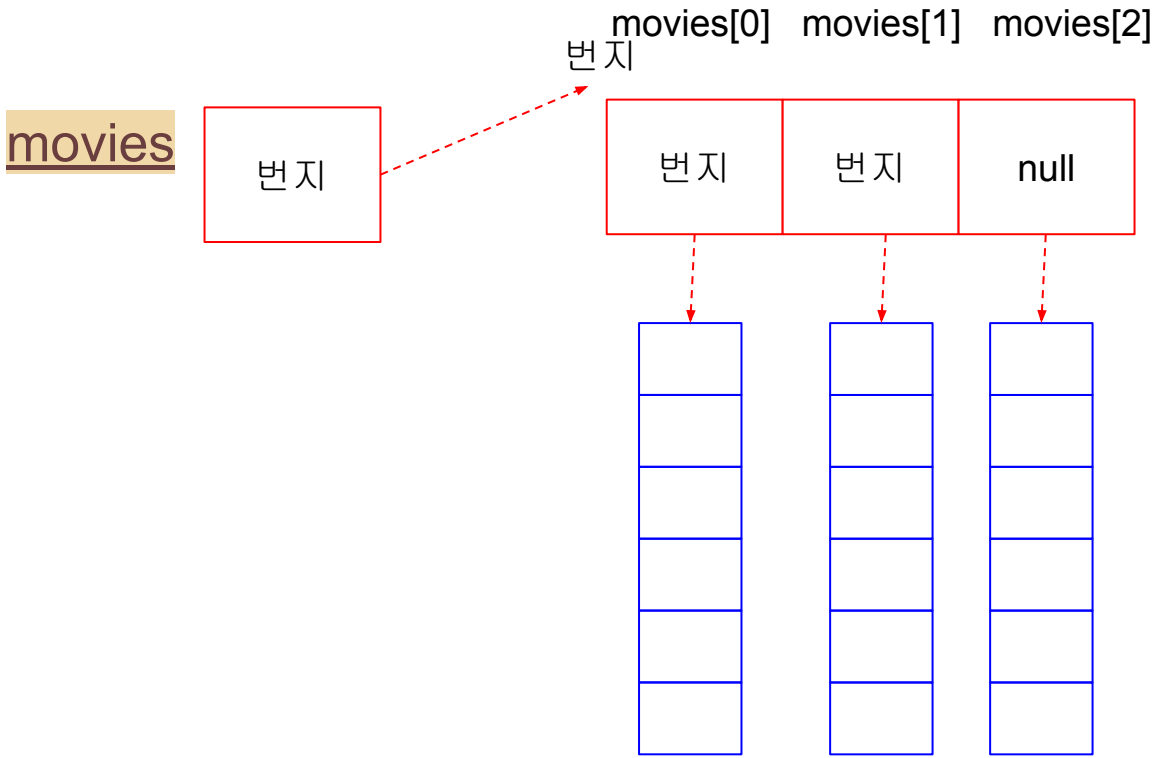
int[ ] int 여러 개 자료형

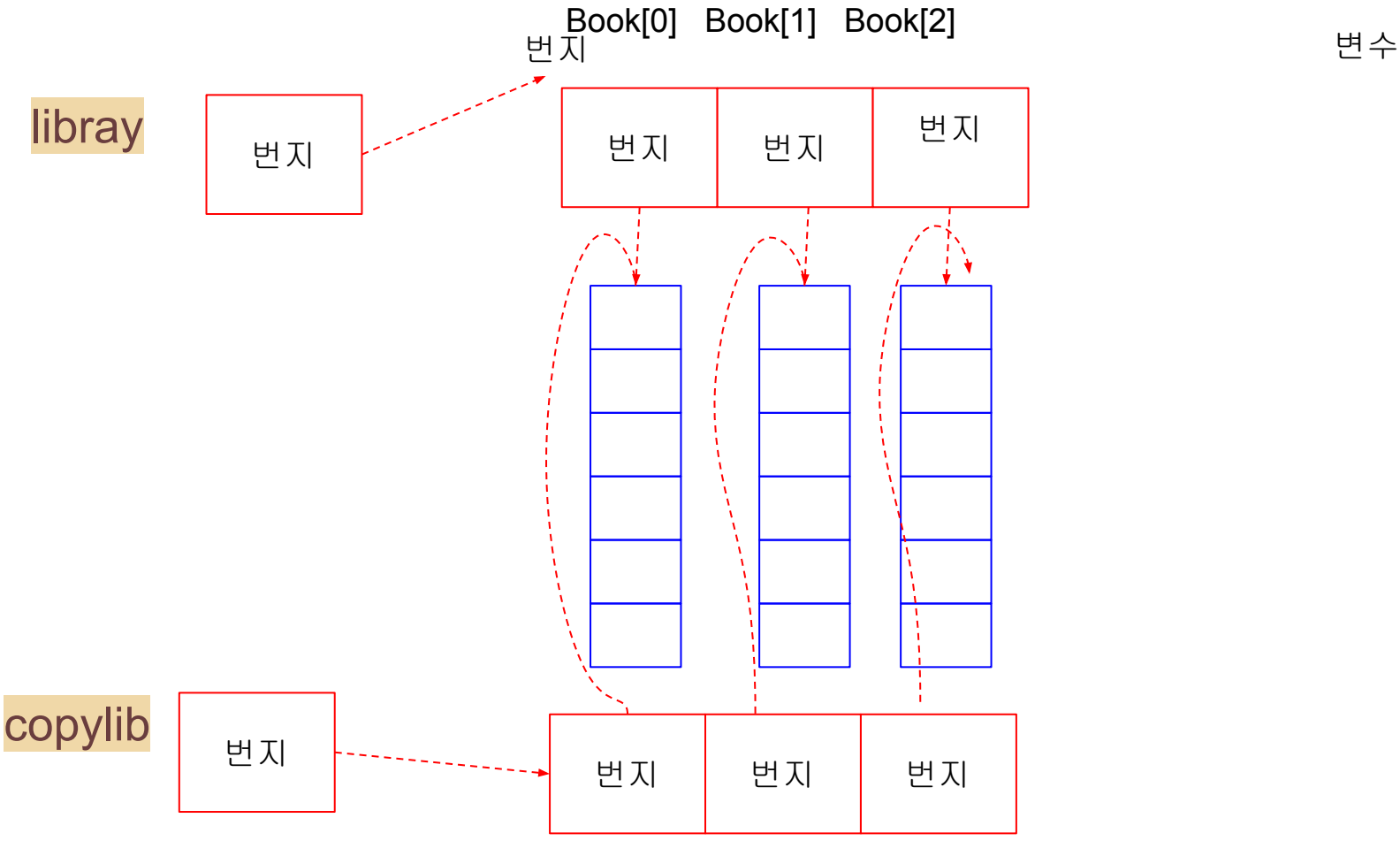
new int[3];

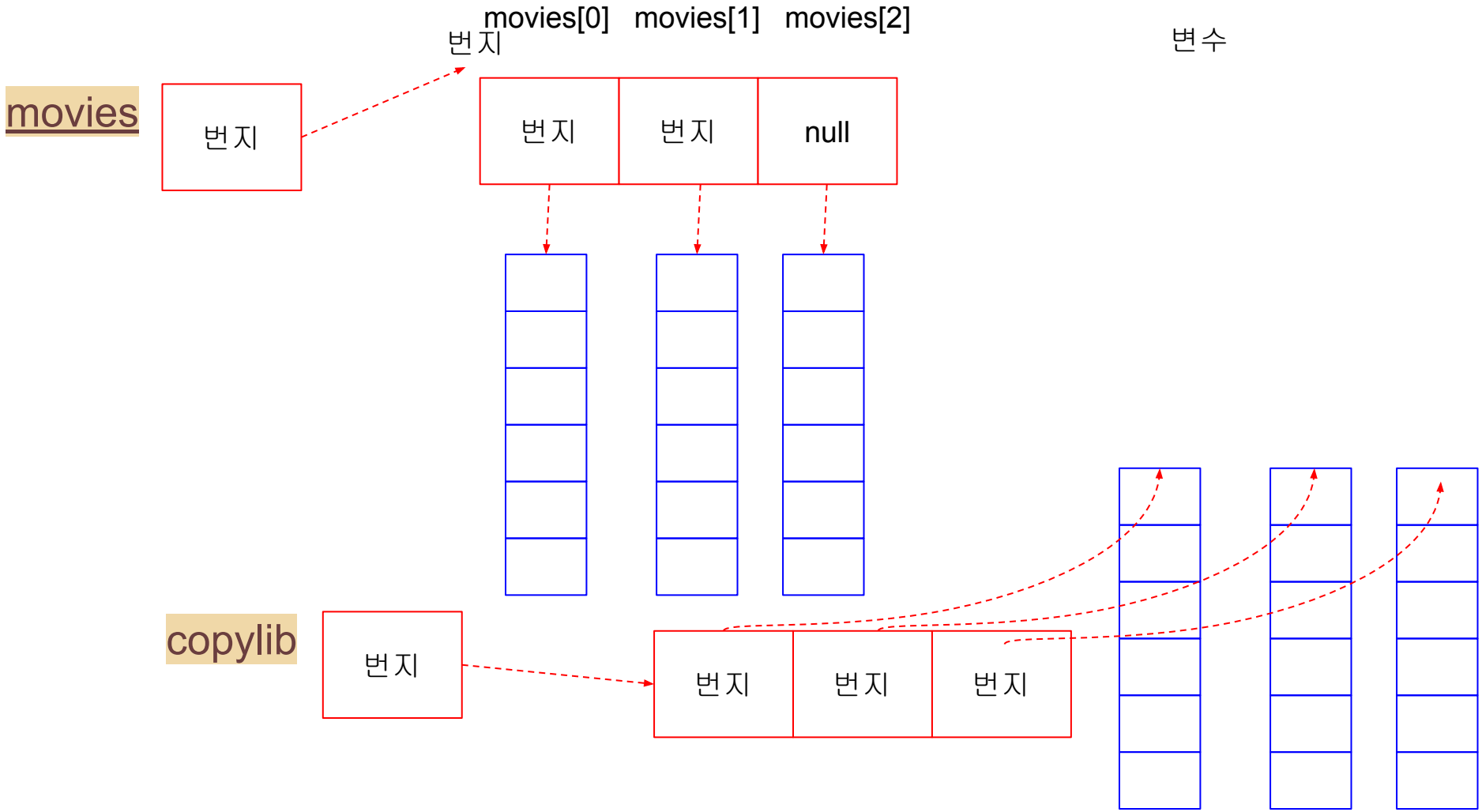
1차원 배열 = [ ]

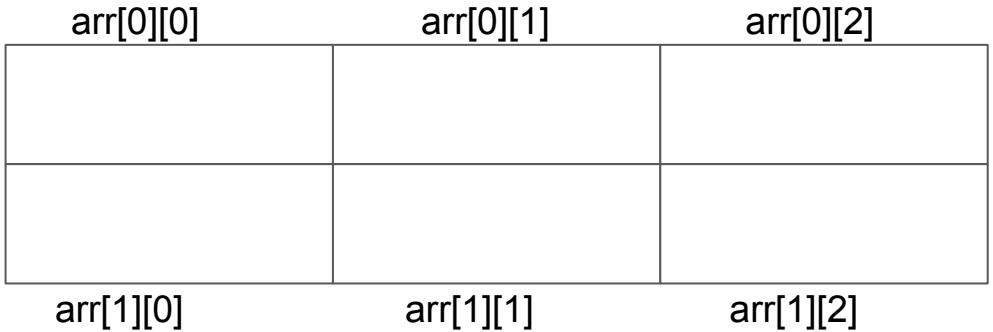
2차원 배열 = [ ][ ]


arr1
번지
번지
0
1
2
3
4
index
arr1[0]
arr1[1]
arr1[2]
arr1[3]
arr1[4]
변수


movies[0]
movies[1]
movies[2]
변수
번지
movies
번지
번지
번지
null

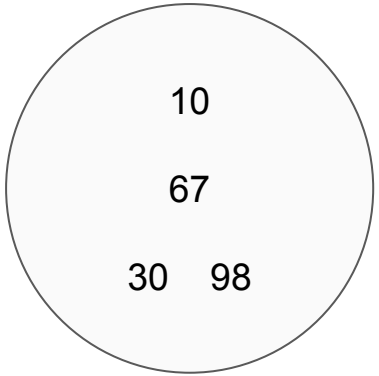

Book[0] Book[1] Book[2]
번지
변수
libray
번지
번지
번지
번지
copylib
번지
번지
번지
번지


movies[0]
movies[1]
movies[2]
번지
변수
movies
번지
번지
번지
null
copylib
번지
번지
번지
번지


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

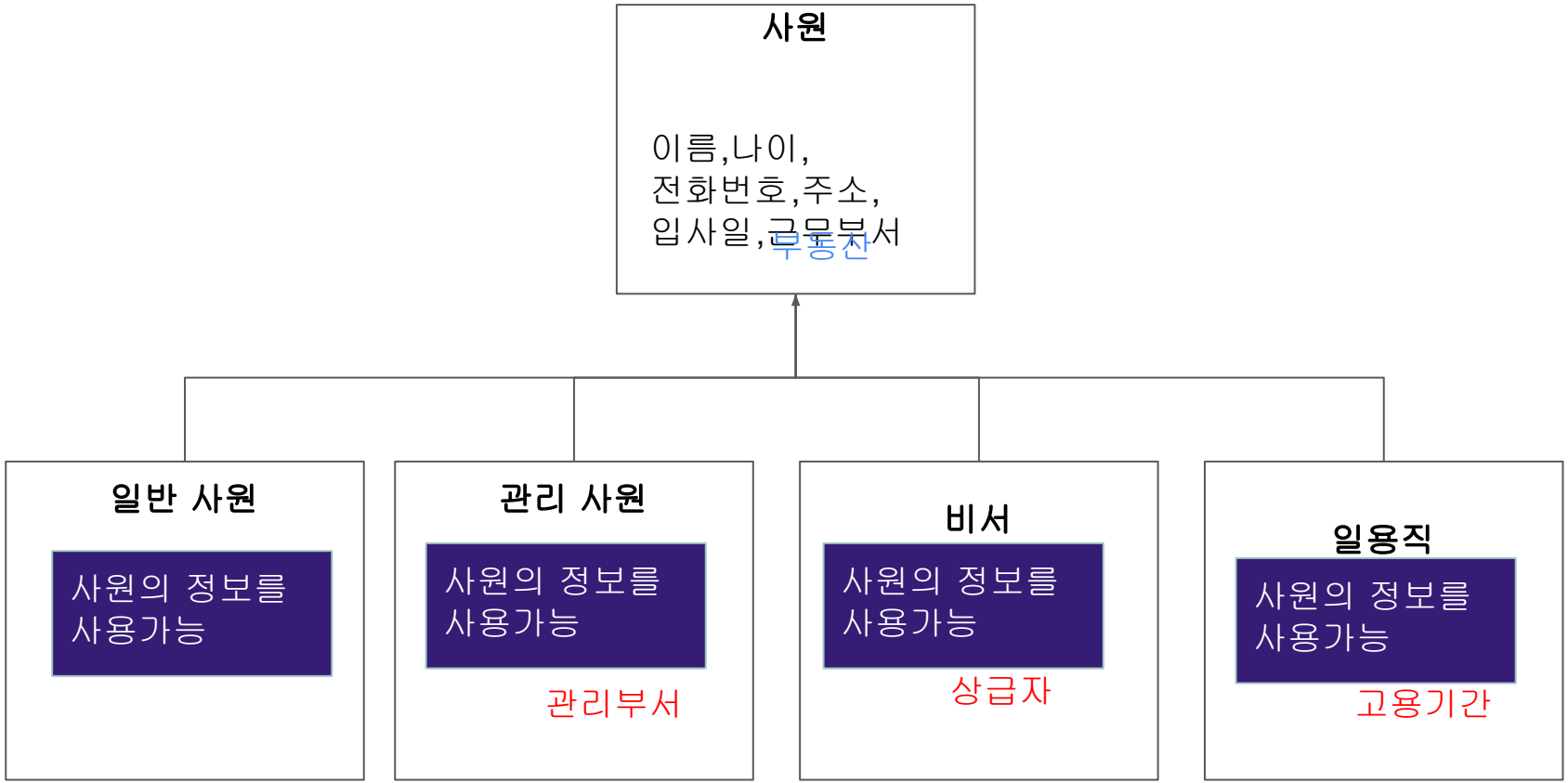
이름,나이,
전화번호,주소,
입사일,관리부서
부동산

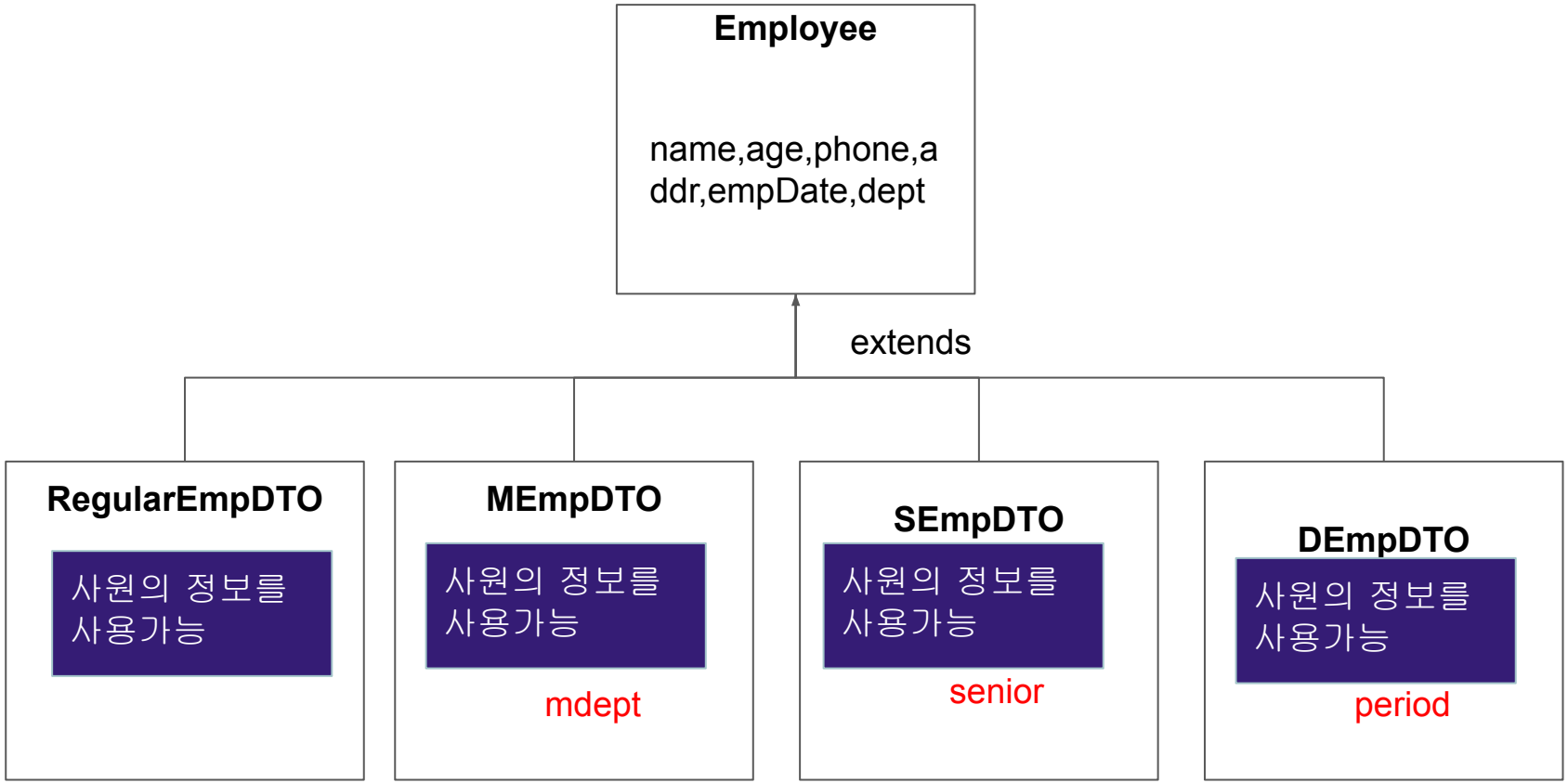
비서

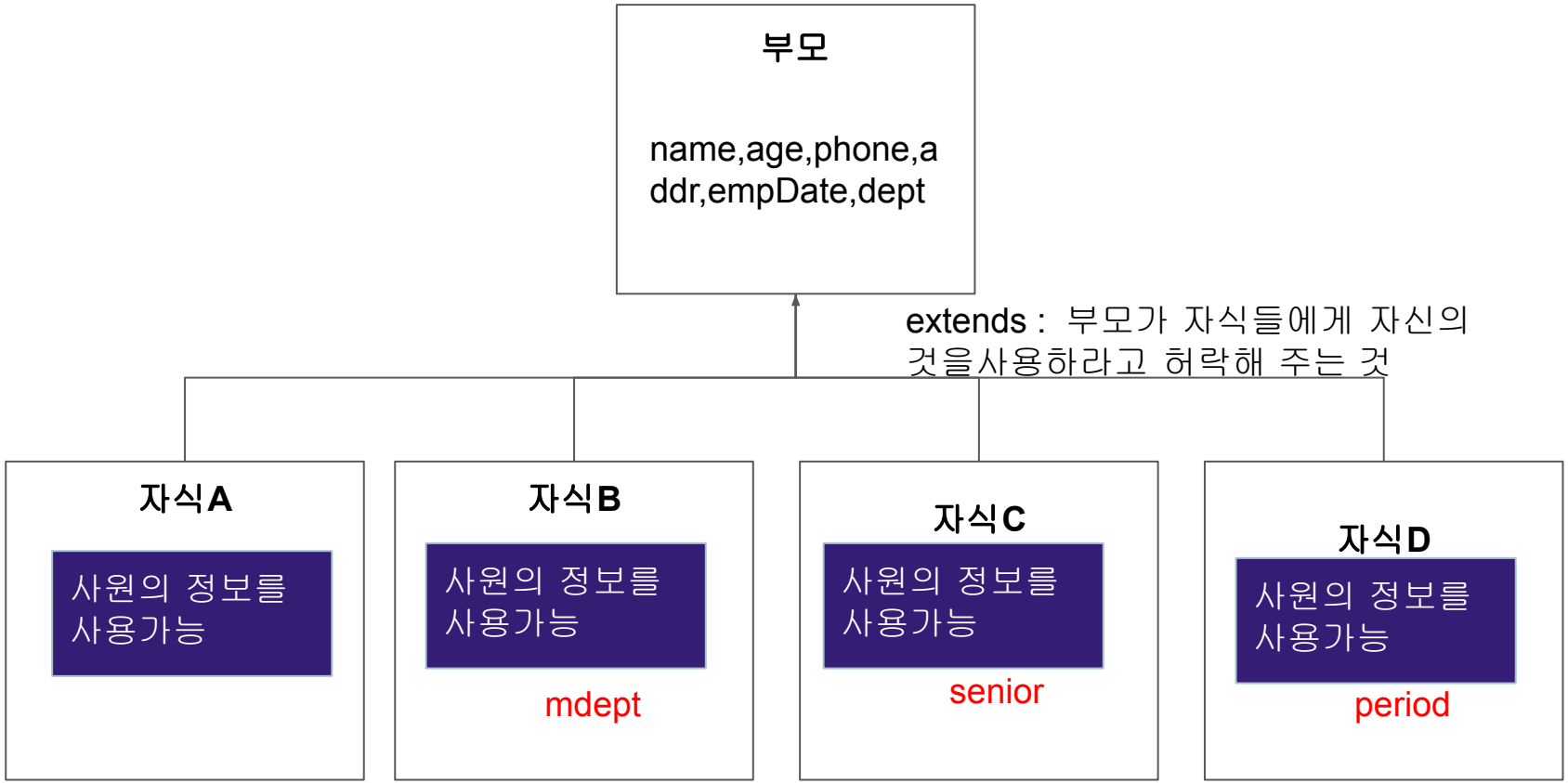
이름,나이,
전화번호,주소,
입사일,상급자
부동산

일용직

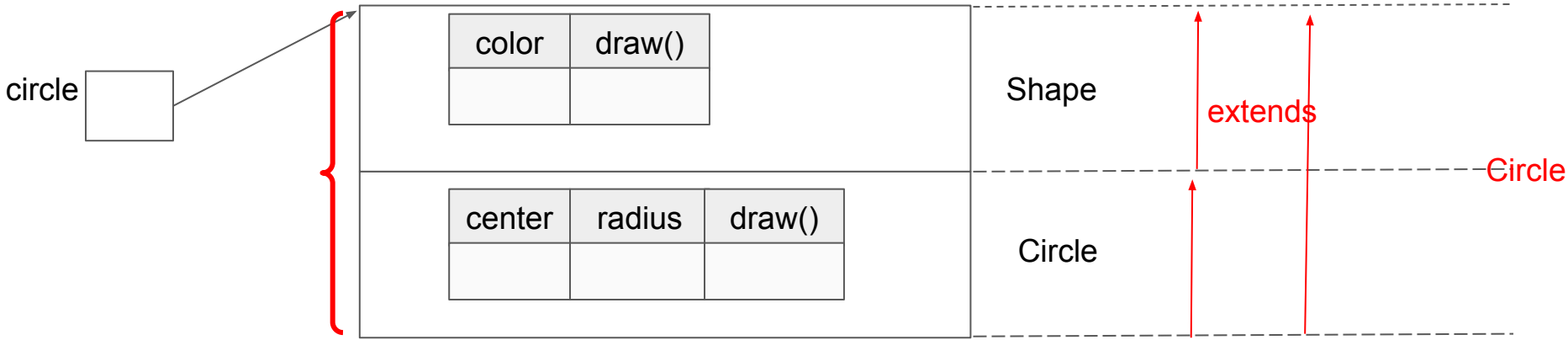
이름,나이,
전화번호,주소,
입사일,고용기간
부동산


사원
이름,나이,
전화번호,주소,
입사일,근무부서
부동산
일반 사원
사원의 정보를
사용가능
관리 사원
사원의 정보를
사용가능
관리부서
비서
사원의 정보를
사용가능
상급자
일용직
사원의 정보를
사용가능
고용기간

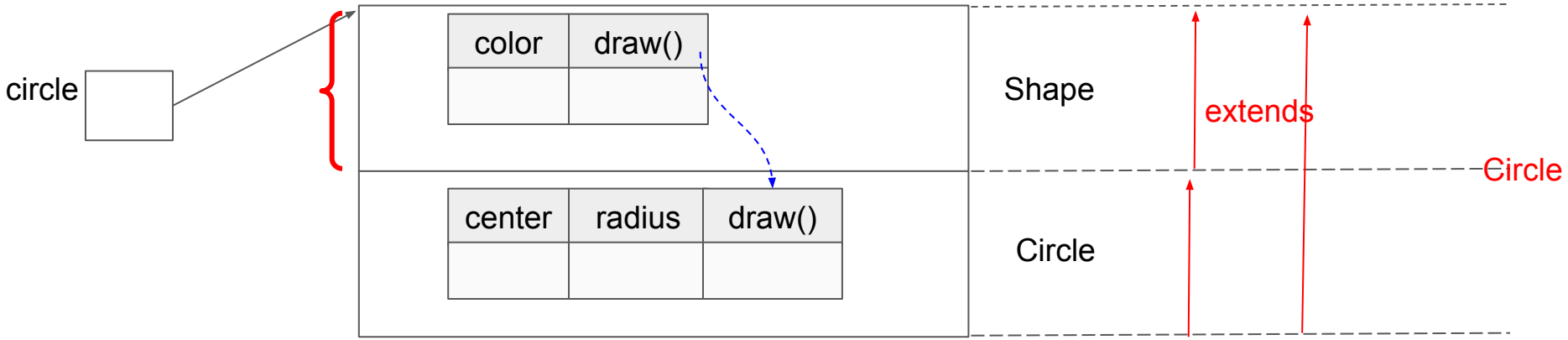

Employee
name,age,phone,addr,empDate,dept
extends
RegularEmpDTO
사원의 정보를 사용가능
MEmpDTO
사원의 정보를 사용가능
mdept
SEmpDTO
사원의 정보를 사용가능
senior
DEmpDTO
사원의 정보를 사용가능
period


부모
name,age,phone,addr,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();
```

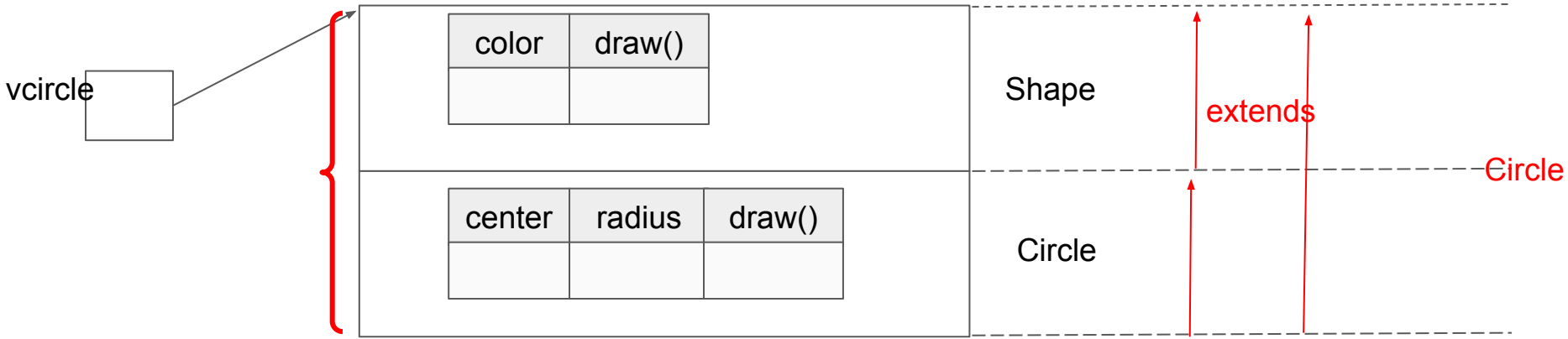


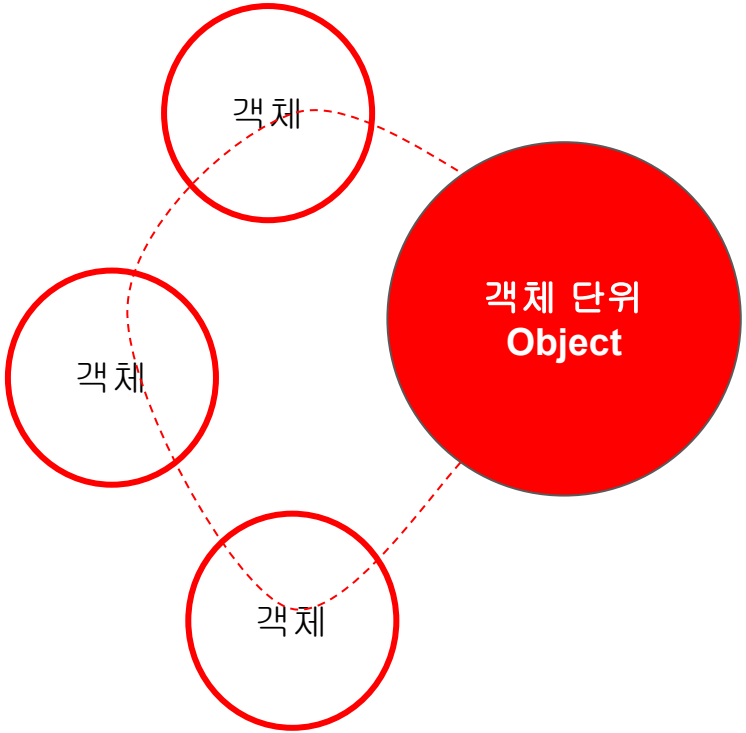
```
Shape circle = new Circle();
```

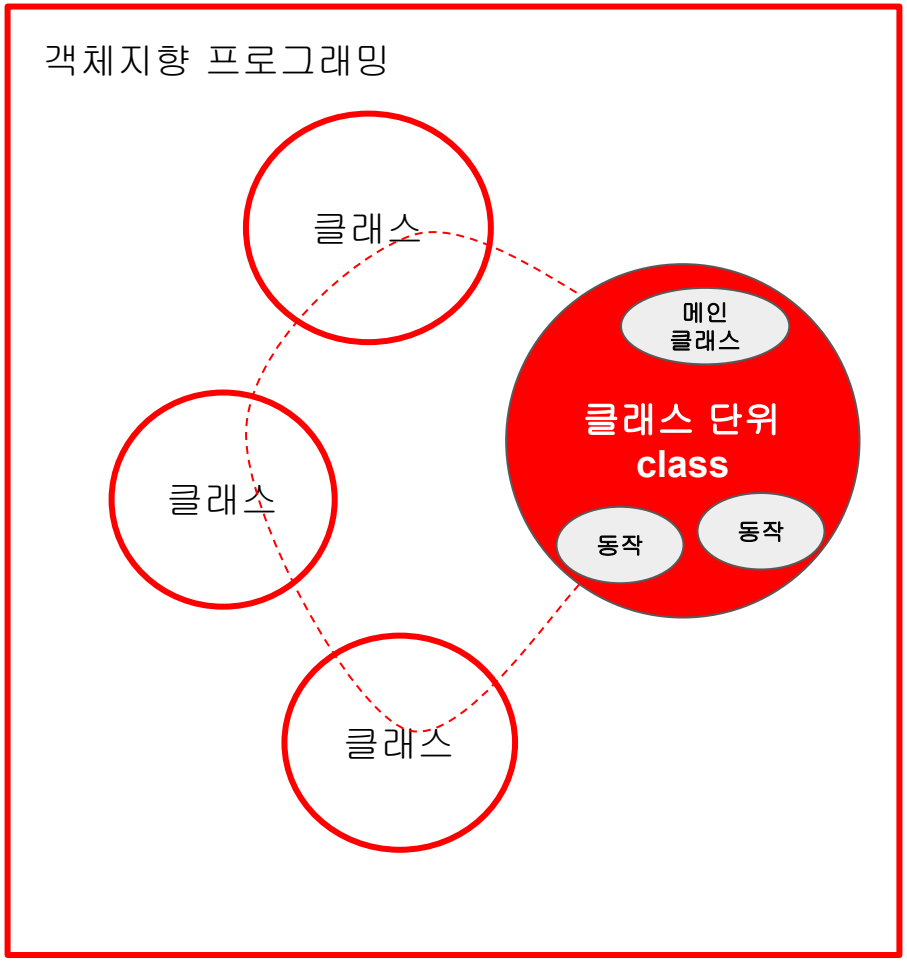
upcasting

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

downcasting

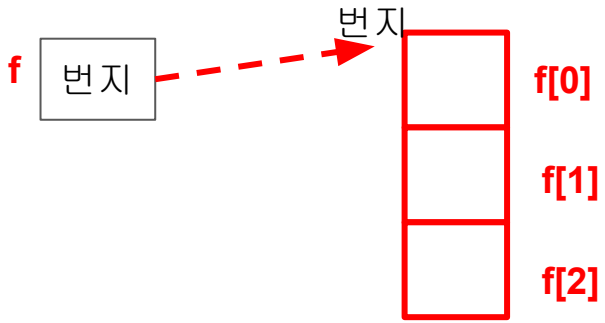



객체
객체
객체
객체 단위
Object


객체지향 프로그래밍
클래스
클래스
클래스
메인
클래스
클래스 단위
class
동작
동작

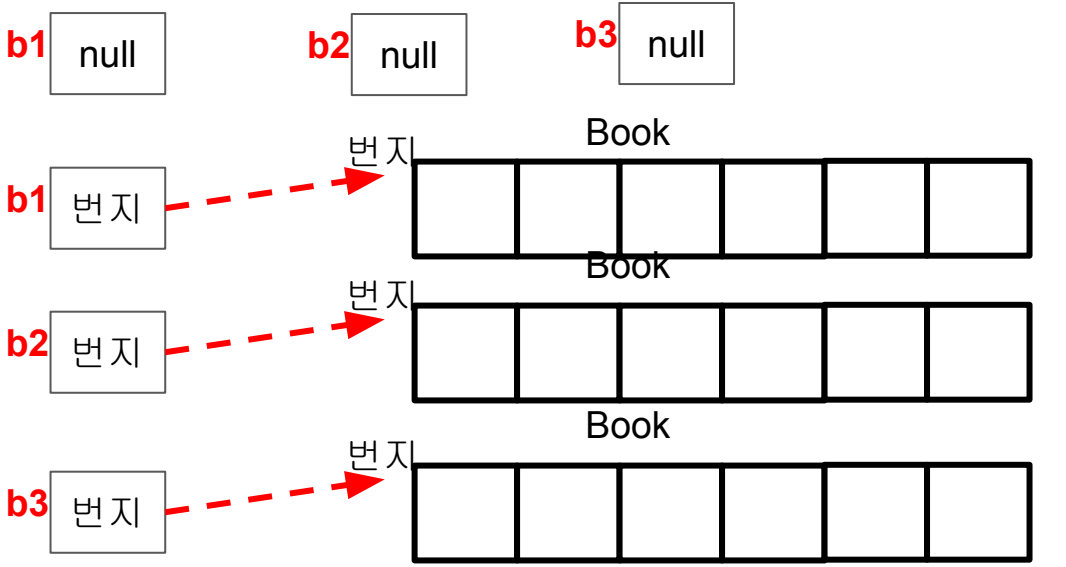
기본배열

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



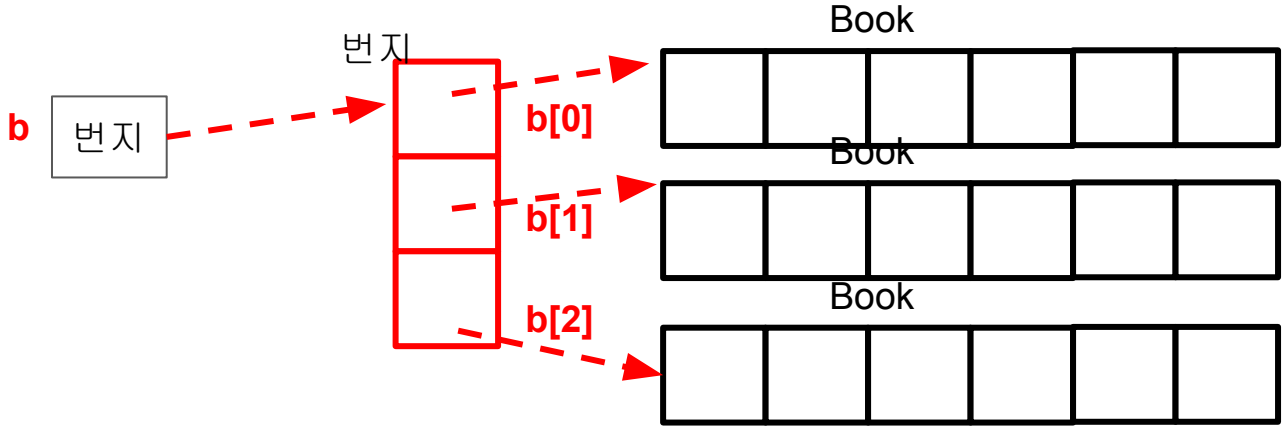
객체

```
Book b1, b2, b3;

b1 = new Book();
b2 = new Book();
b3 = new Book();
```



객체배열

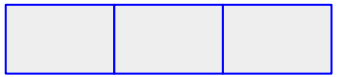
```
Book[ ] b = new Book[3];

b[0] = new Book();
b[1] = new Book();
b[2] = new Book();
```



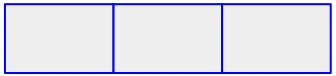
2차원 배열

1차원 배열을 3개 생성

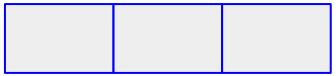
int[ ] kor = new int[3];



int[ ] eng = new int[3];

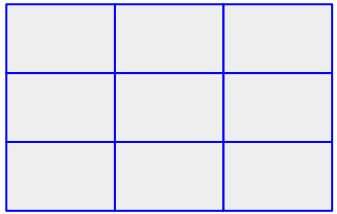


int[ ] mat = new int[3];



2차원 배열을 1개 생성 (이미지)

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



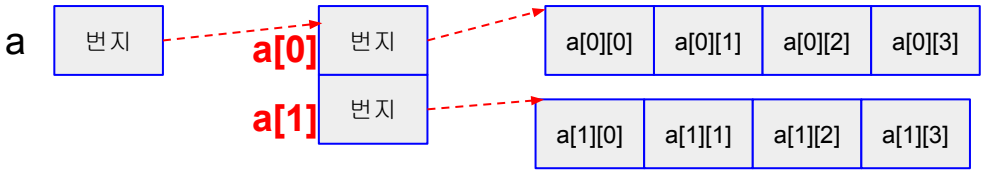
2차원 배열 생성

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

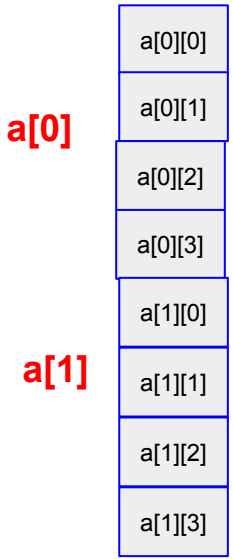
4열

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]

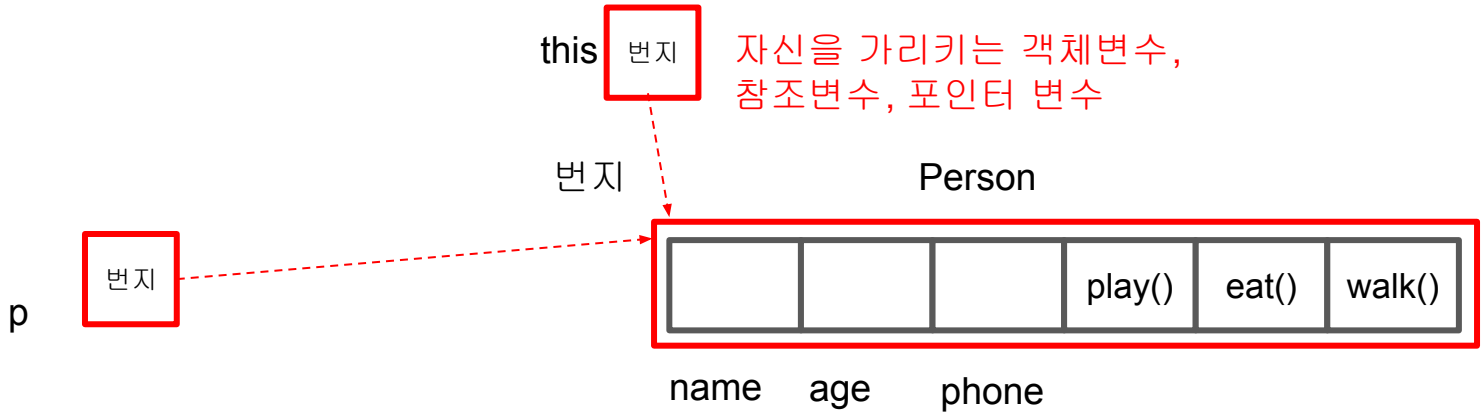


2차원 배열도 연속적인 자료구조

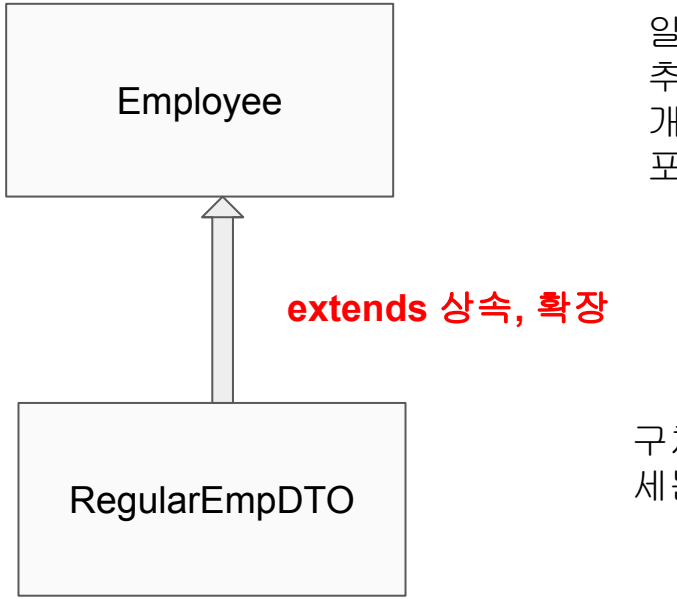


Person p = new Person();

생성자메서드



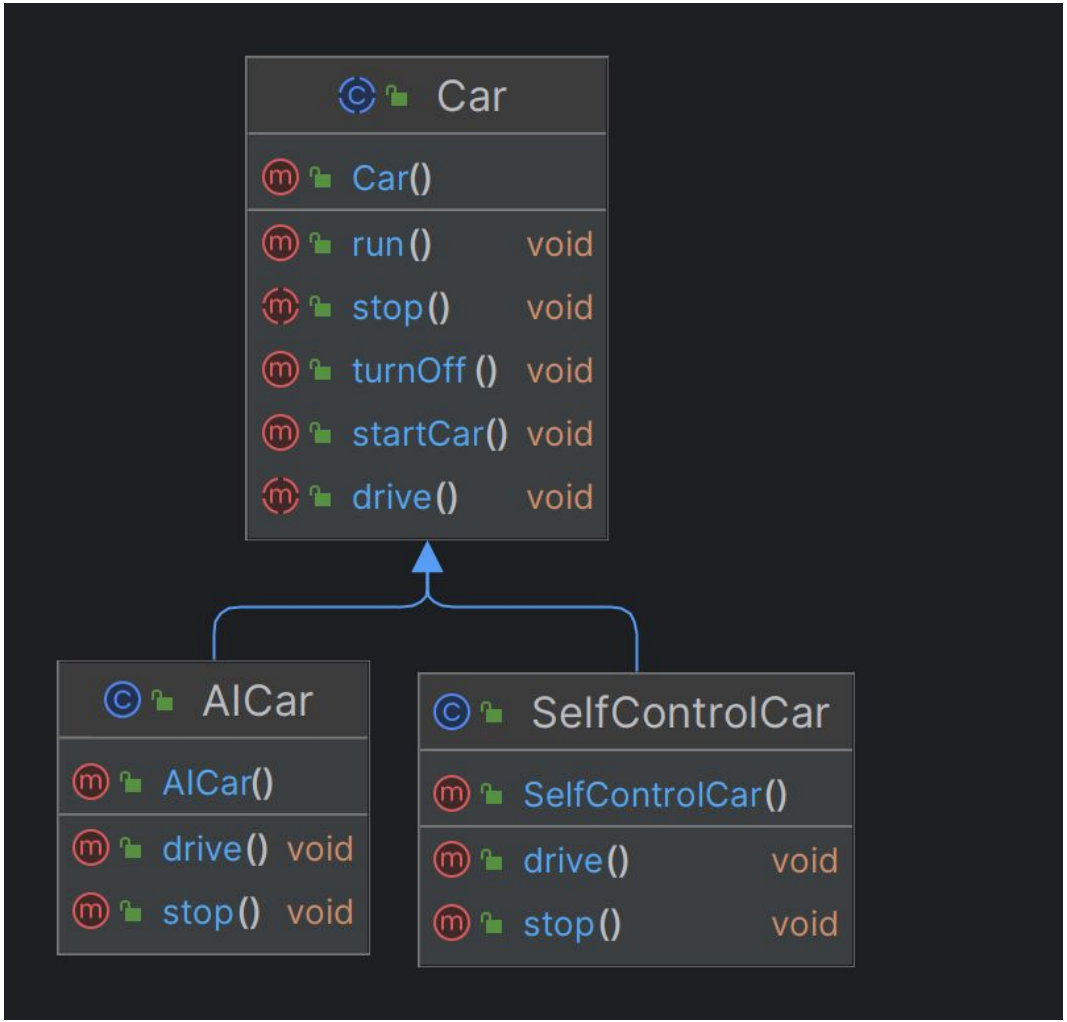
super class
상위 클래스
부모 클래스

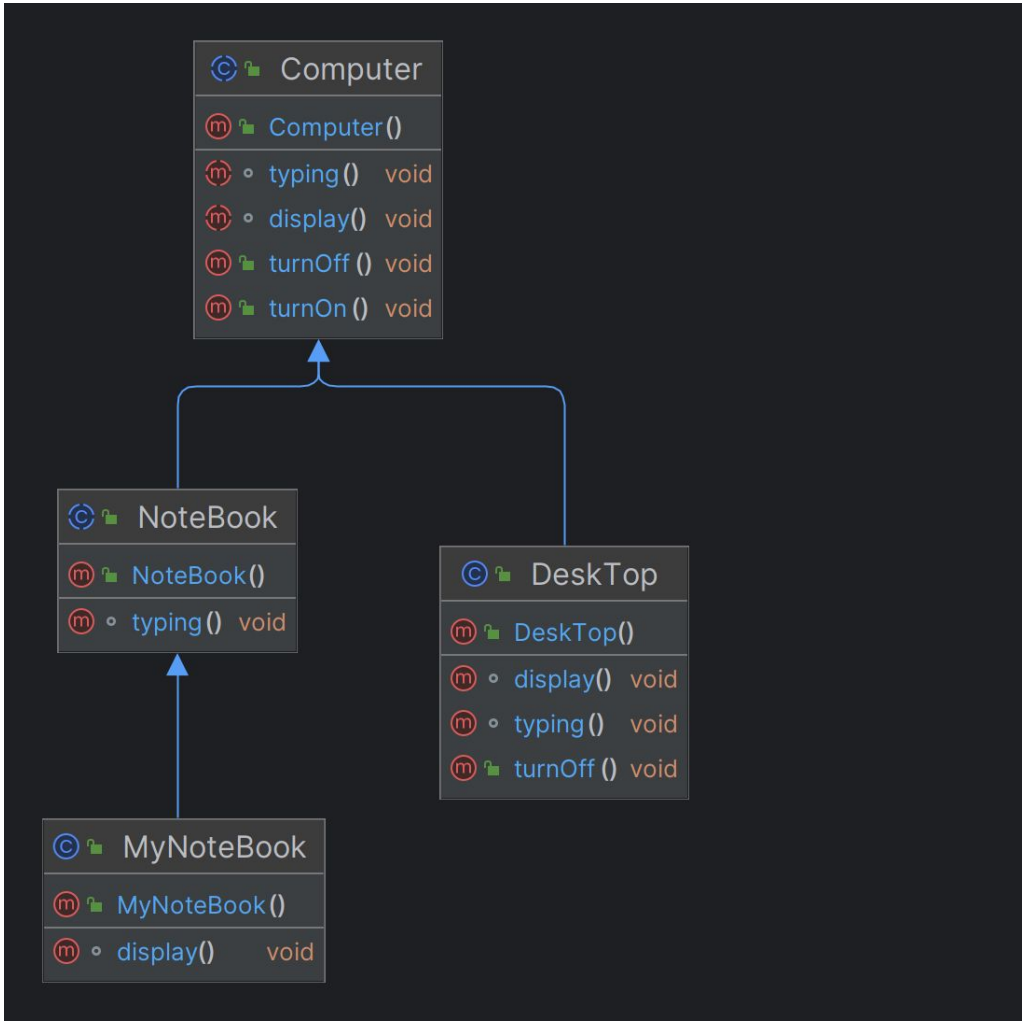


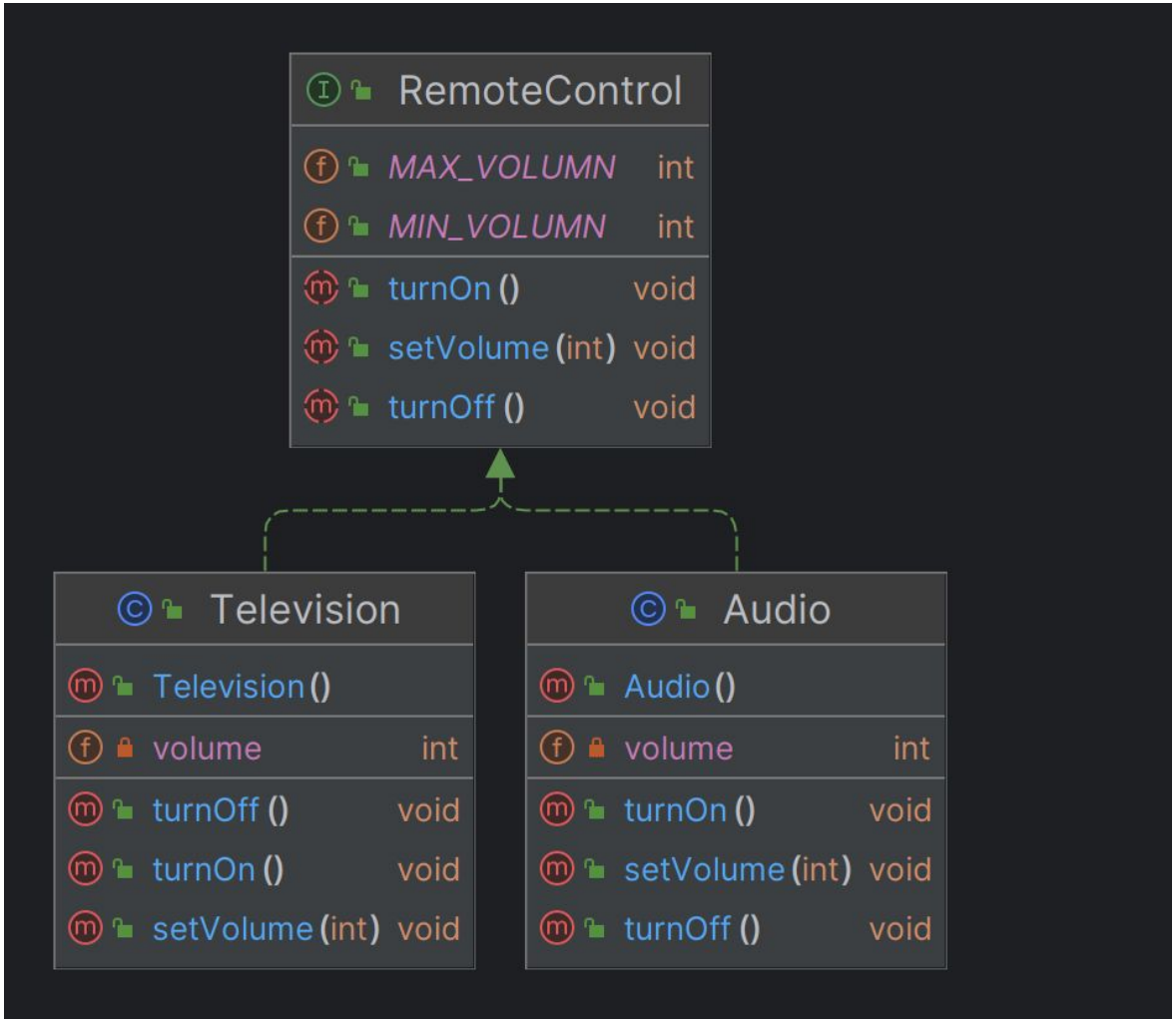
일반화
추상화
개념화
포괄적

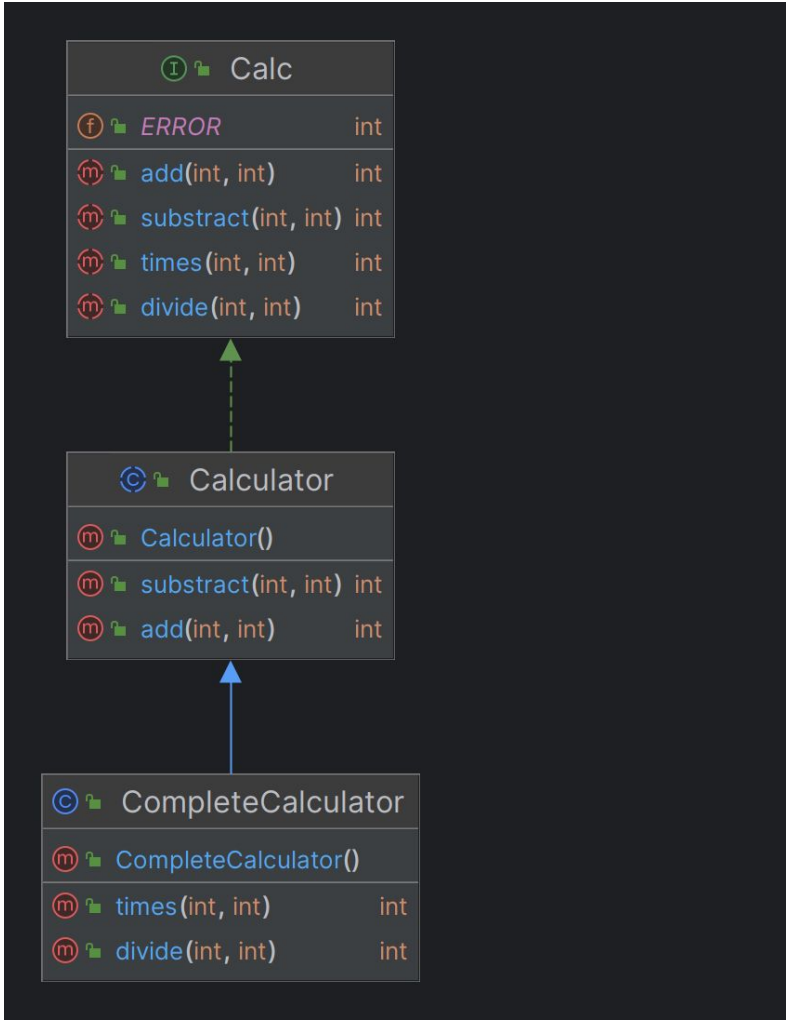
sub class
하위 클래스
자식 클래스
파생 클래스

구체화
세분화

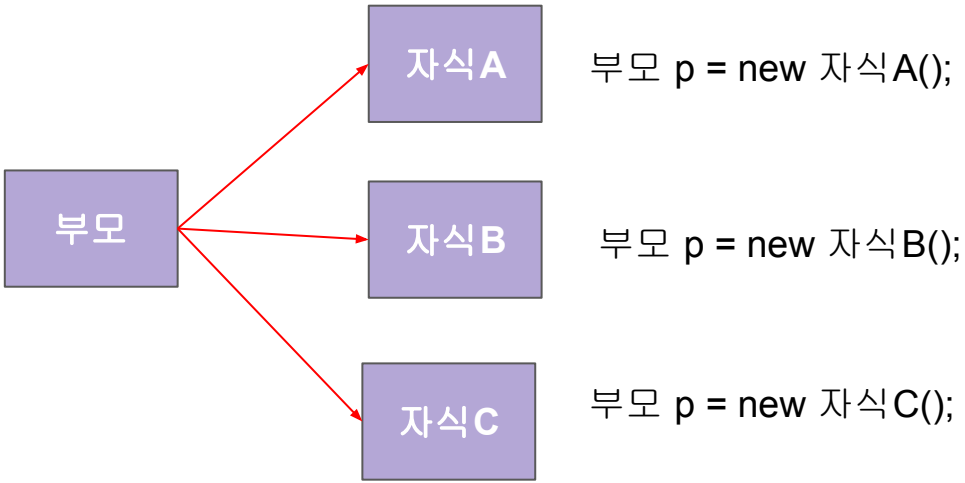

Car
Car()
run() void
stop() void
turnOff() void
startCar() void
drive() void
AICar
AICar()
drive() void
stop() void
SelfControlCar
SelfControlCar()
drive() void
stop() void


Computer
Computer()
typing() void
display() void
turnOff() void
turnOn() void
NoteBook
NoteBook()
typing() void
DeskTop
DeskTop()
display() void
typing() void
turnOff() void
MyNoteBook
MyNoteBook()
display() void


RemoteControl
MAX_VOLUMN int
MIN_VOLUMN int
turnOn () void
setVolume (int) void
turnOff () void
Television
Television ()
volume int
turnOff () void
turnOn () void
setVolume (int) void
Audio
Audio ()
volume int
turnOn () void
setVolume (int) void
turnOff () void


Calc
ERROR int
add(int, int) int
substract(int, int) int
times(int, int) int
divide(int, int) int
Calculator
Calculator()
substract(int, int) int
add(int, int) int
CompleteCalculator
CompleteCalculator()
times(int, int) int
divide(int, int) int

Upcasting

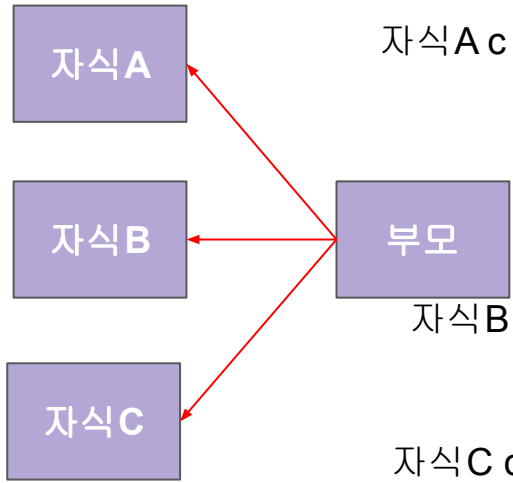


부모 p = new 자식A();

부모 p = new 자식B();

부모 p = new 자식C();

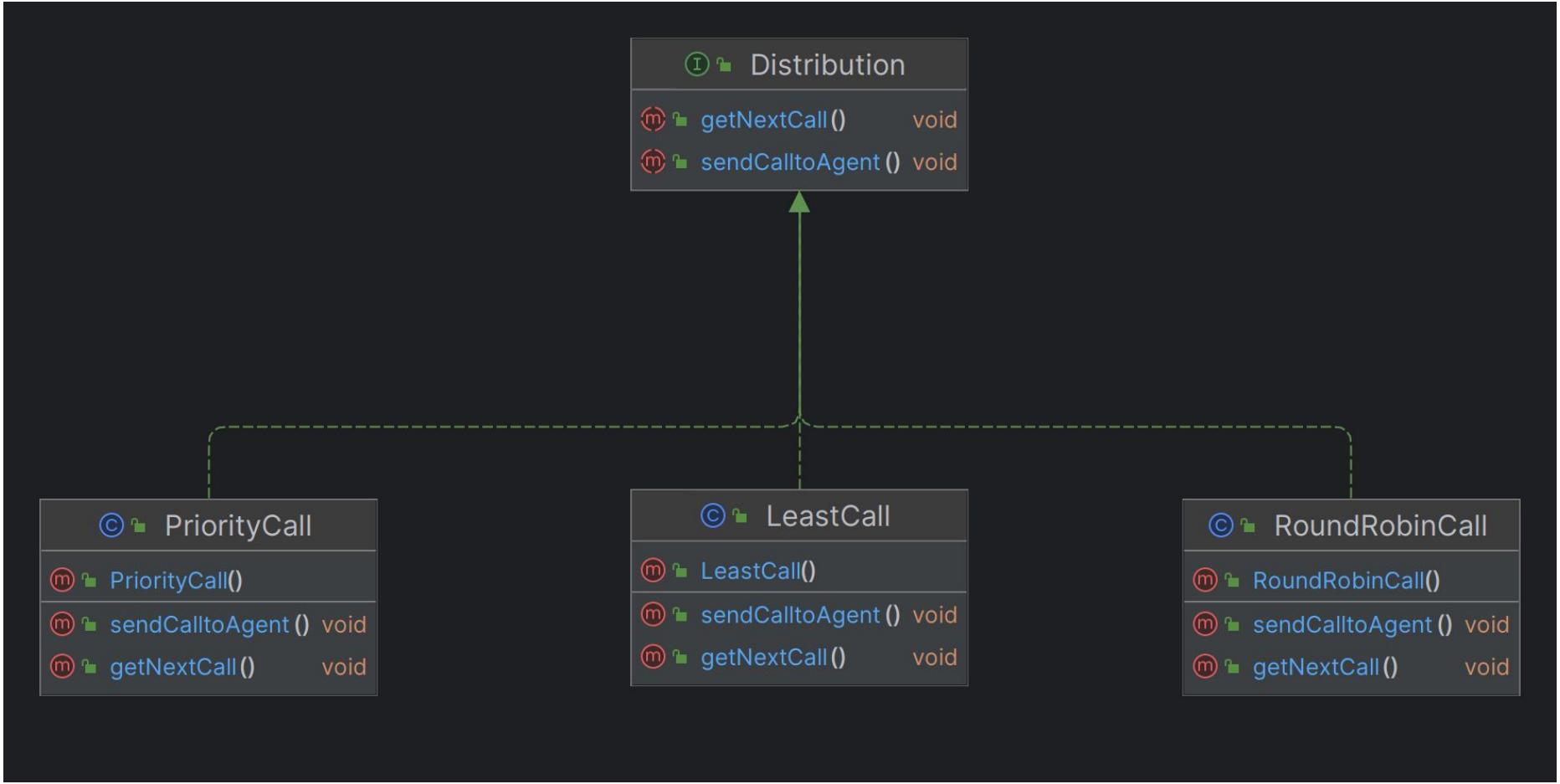
Downcasting

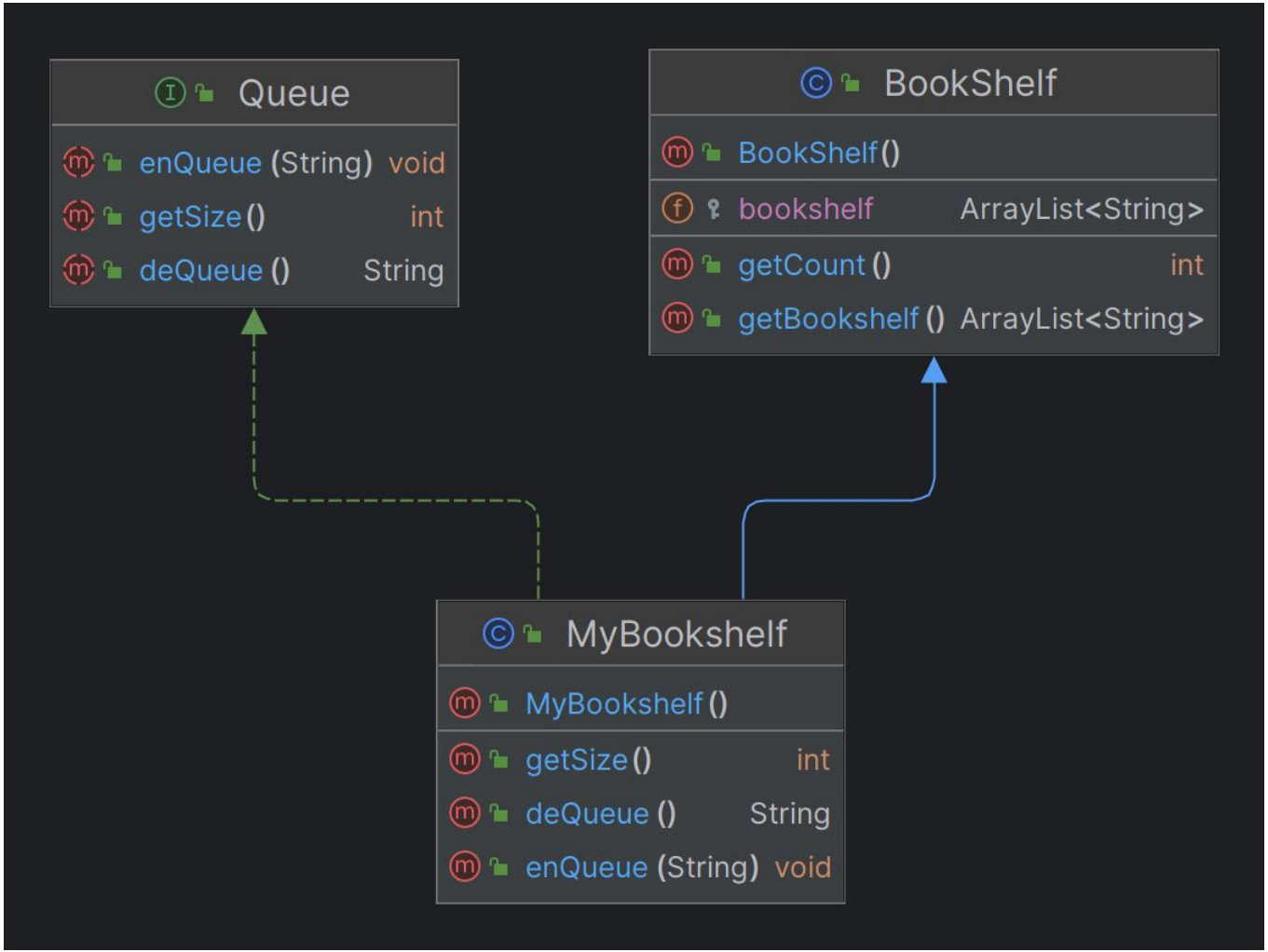


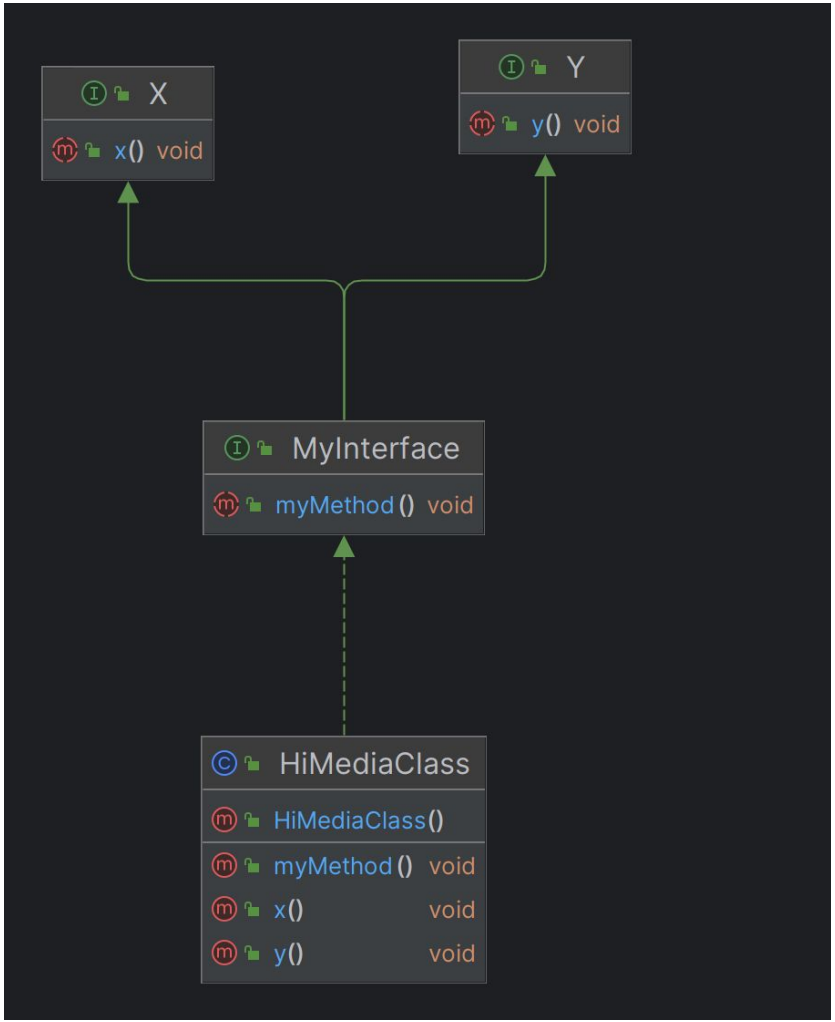
자식A c = (자식A)p;

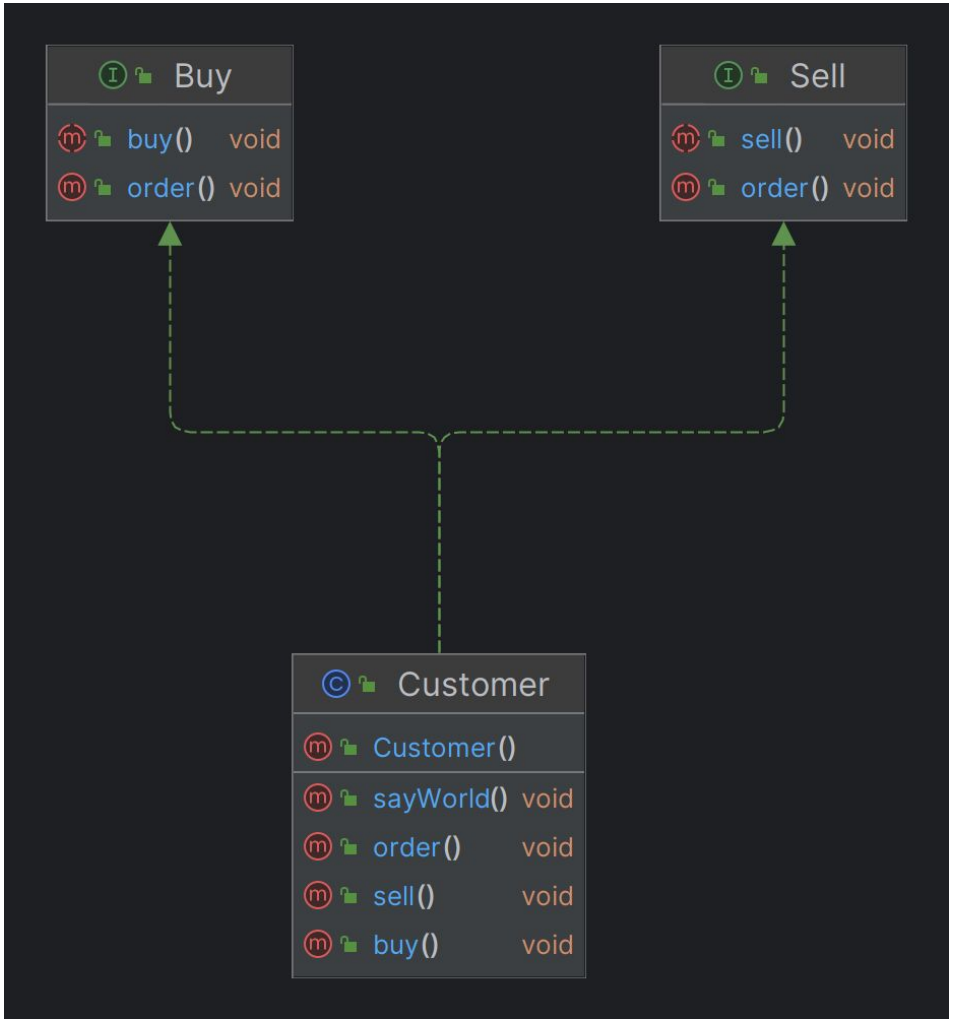
자식B c = (자식B)p;

자식C c = (자식C)p;


Distribution
getNextCall() void
sendCalltoAgent() void
PriorityCall
PriorityCall()
sendCalltoAgent() void
getNextCall() void
LeastCall
LeastCall()
sendCalltoAgent() void
getNextCall() void
RoundRobinCall
RoundRobinCall()
sendCalltoAgent() void
getNextCall() void


Queue
enQueue (String) void
getSize () int
deQueue () String
BookShelf
BookShelf ()
bookshelf ArrayList<String>
getCount () int
getBookshelf () ArrayList<String>
MyBookshelf
MyBookshelf ()
getSize () int
deQueue () String
enQueue (String) void


X
x() void
Y
y() void
MyInterface
myMethod() void
HiMediaClass
HiMediaClass()
myMethod() void
x() void
y() void


Buy
buy() void
order() void
Sell
sell() void
order() void
Customer
Customer()
sayWorld() void
order() void
sell() void
buy() void