CS study 17주차

정리

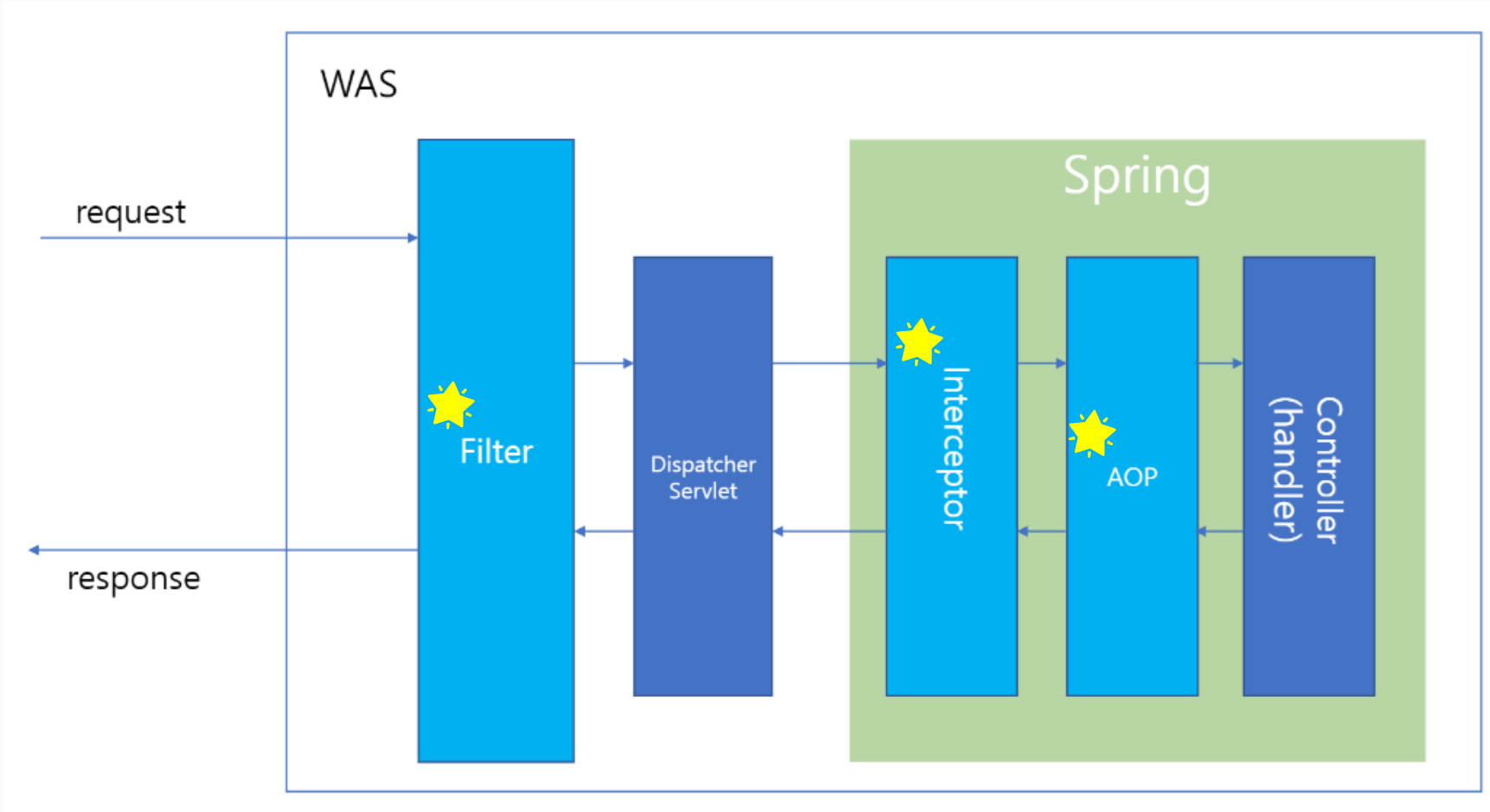
Spring AOP, 인터셉터, 필터 / Bean의 생명주기

JAVA JVM 메모리 구조와 가비지

SW Rest, RESTful, REST API / SSR과 CSR / webSocket / TDD

DB index

Spring – 필터, 인터셉터, AOP



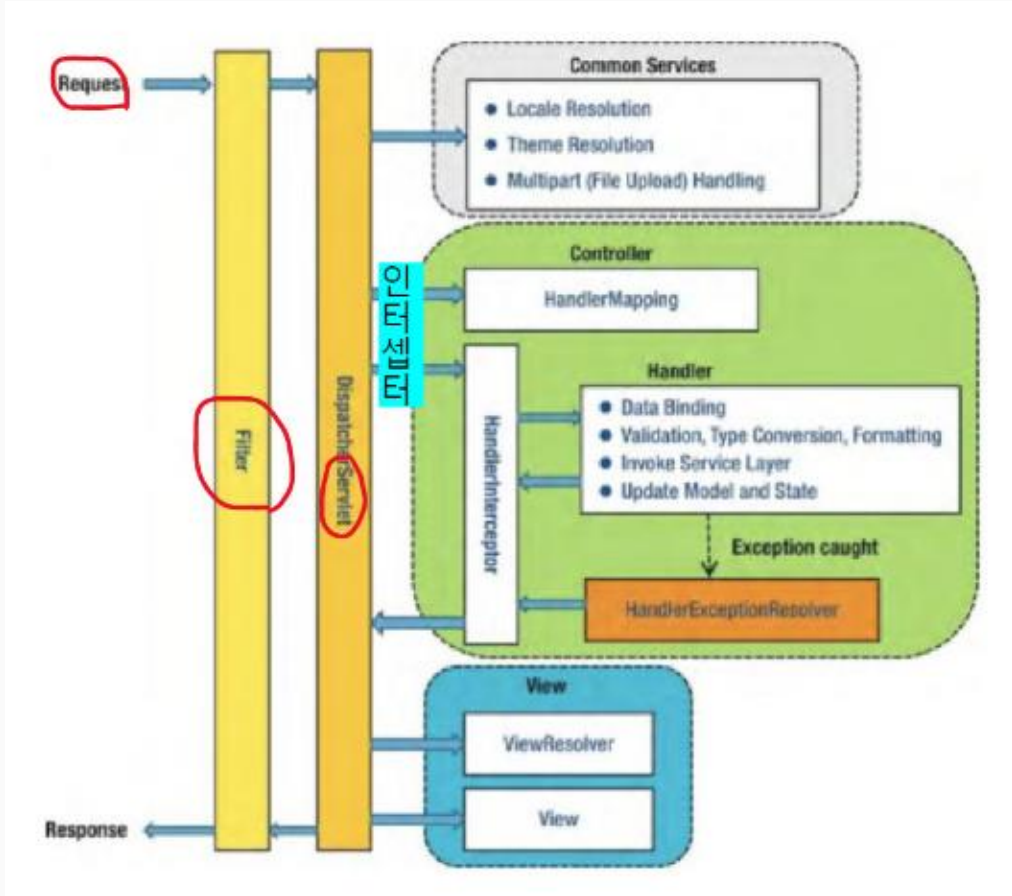
Filter란?

클라이언트로부터 오는 요청과 Servlet 사이에 위치

클라이언트의 요청 정보와 요청 결과를 알맞게 변경

Servlet Request/Reponse에 대한 사전/사후 처리 가능

Spring의 독자적인 기능이 아닌 Servlet에서 제공



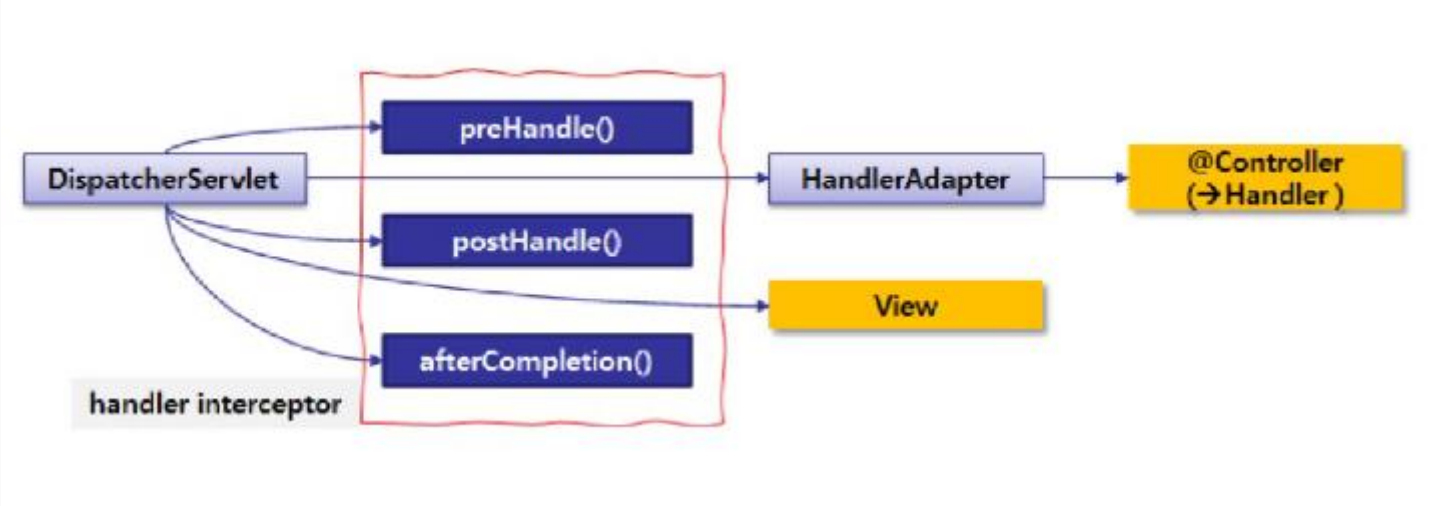
Filter란?

Filter chain을 통해 여러 필터가 연쇄적으로 동작

요청에 대한 인증, 권한 체크 (ex request 헤더를 검사해 JWT 토큰 검사 등)

인코딩 설정

Interceptor란?



클라이언트에서 서버로 들어온 controller로 가는 요청을 가로챔

여러 컨트롤러에서 공통적으로 사용되는 기능을 정의

컨트롤러의 수정 없이 컨트롤러 수행 전/후처리 동작을 추가해 컨트롤러의 반복적인 코드 제거

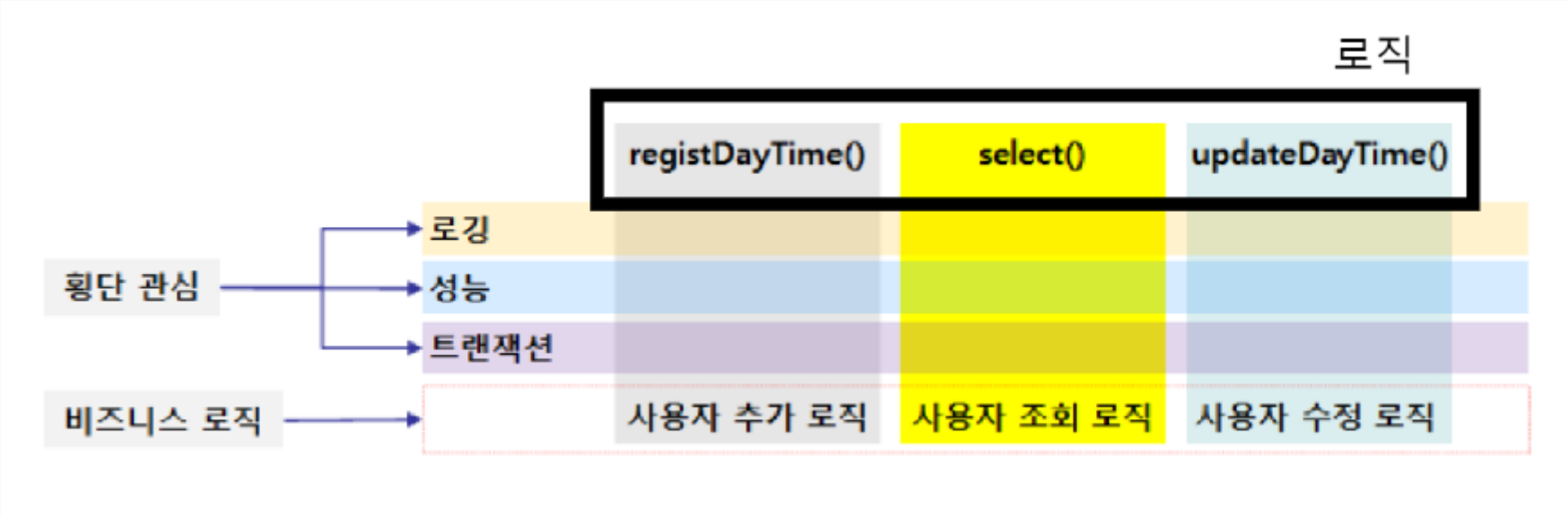
Interceptor란?

필터와 매우 유사하지만, 인터셉터는 조금 더 세분화된 컨트롤러의 반복 코드 제거나 권한 확인시

사용하는 것을 권장함

필터는 모든 요청과 응답에 관한 처리, 특정 컨텐츠에 필터를 매핑하는 경우 사용됨

AOP란?



관점 지향 프로그래밍

서비스 로직은 비즈니스 로직과 부가기능으로 나눌 수 있는데, 중복되는 부가기능을 모듈화하고 비즈니스 로직에서 분리해 재사용할 수 있게 만드는 것

Filter vs Interceptor vs AOP 공통점

- 공통된 부분을 모듈화 함
- 어떠한 행동을 하기 전 먼저 실행 or 실행 후 추가적인 행동을 할 때 사용되는 기능들

Spring – Spring Bean

- Spring IOC 컨테이너가 관리하는 자바 객체
- 컨테이너? 객체를 생성, 관리, 책임지고 의존성을 관리해줌
- @Component 이용, 직접 등록(@Configuration/@Bean)
- Singleton으로 생성, 관리

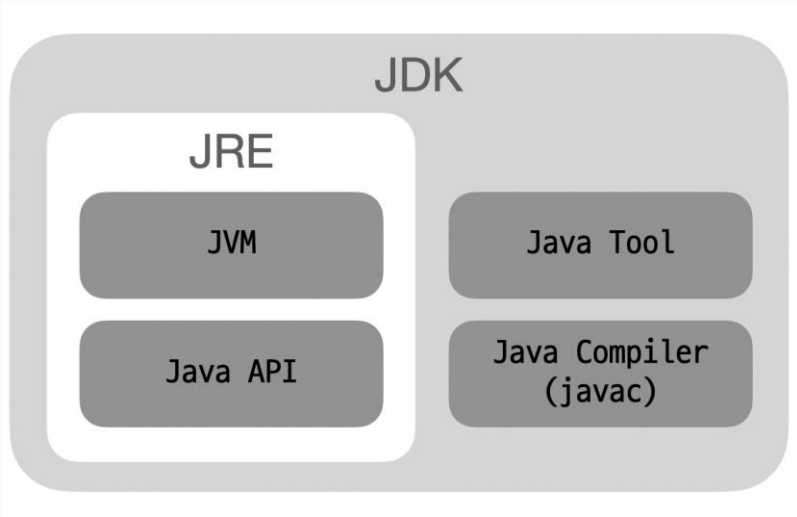
Spring – Spring Bean의 생명주기

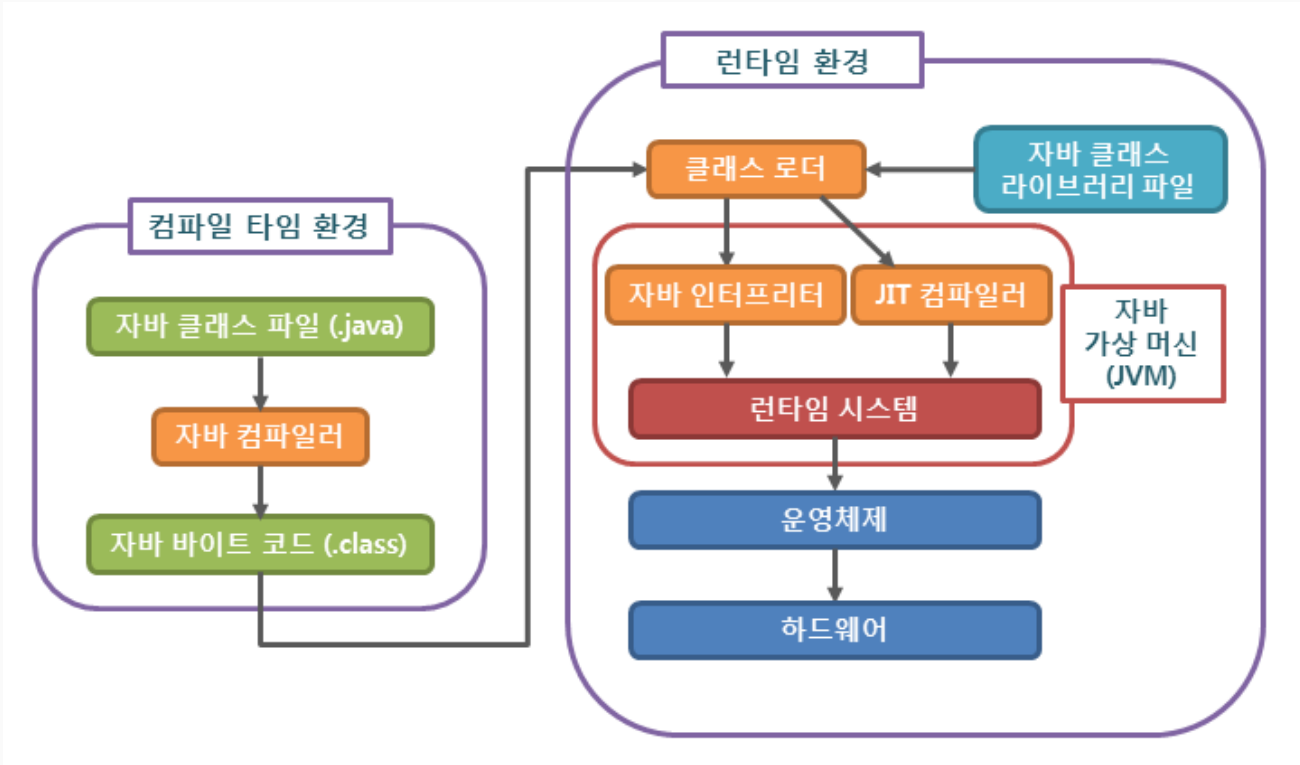
1. 설정 파일을 읽어 Bean 객체 생성
2. Bean 객체에 의존성 주입
3. 빈 생성 생명주기 콜백 (초기화)

--- 어플리케이션에서 빈 사용 ---

4. 빈 소멸 생성주기 콜백

JAVA – JVM

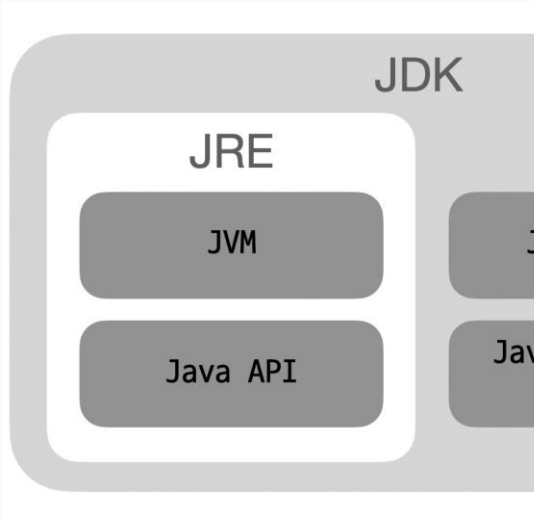


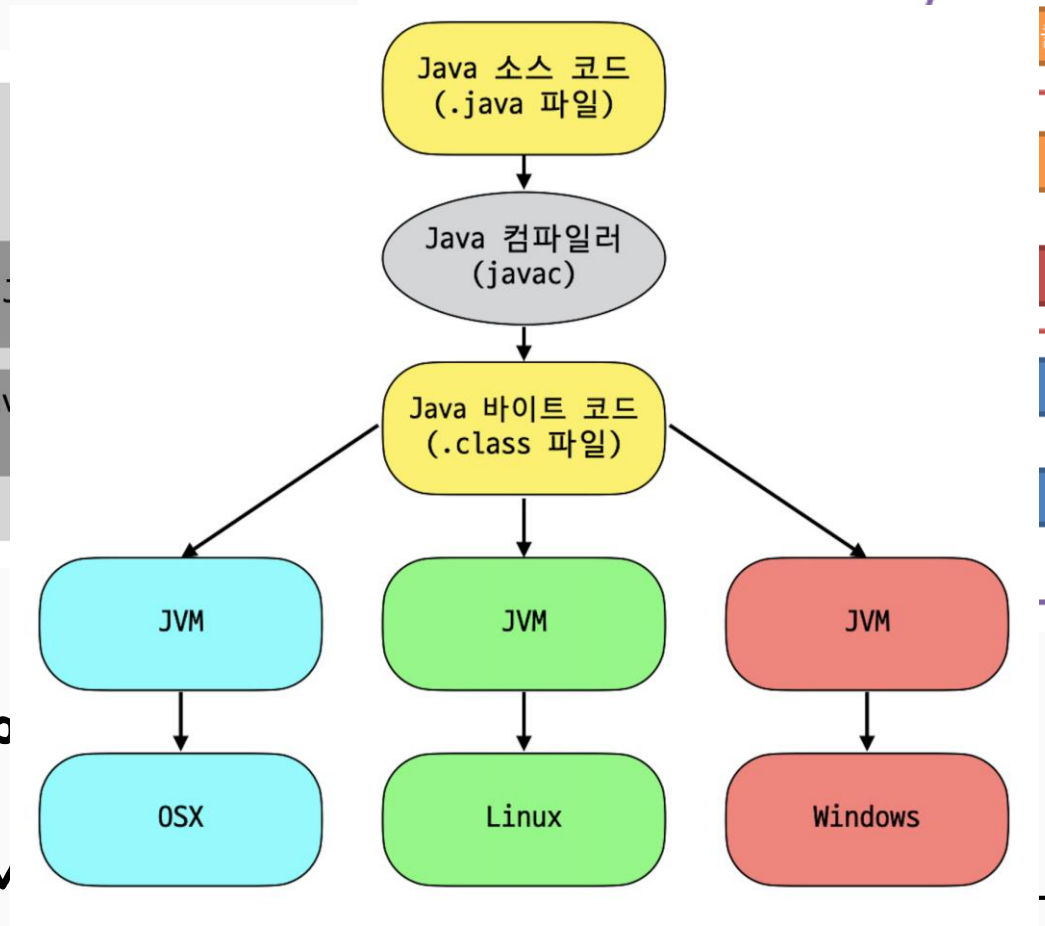


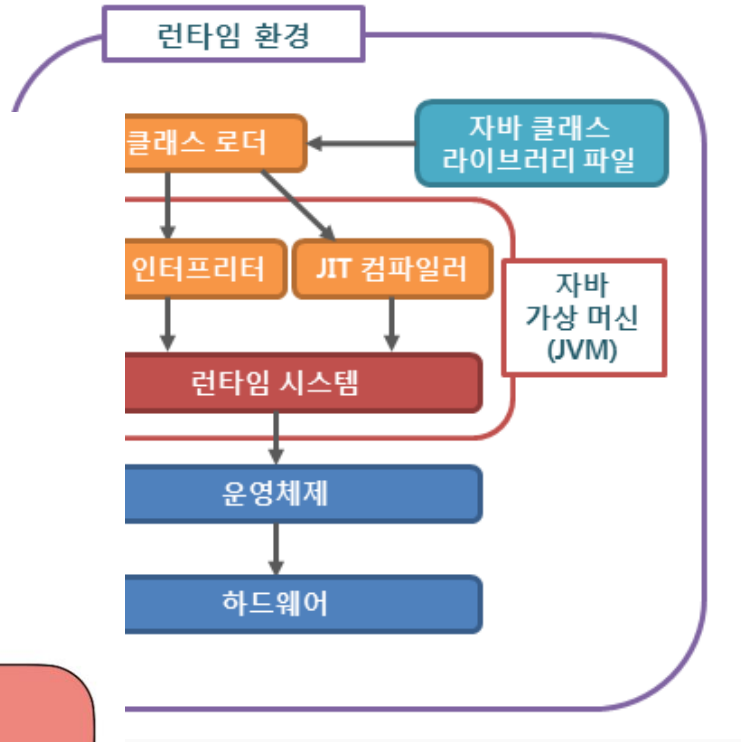
자바 바이트코드는 JRE위에서 동작함

JRE = 자바 API + JVM → JVM = 자바 바이트 코드를 해석하고, 실행하는 가상머신

JAVA – JVM







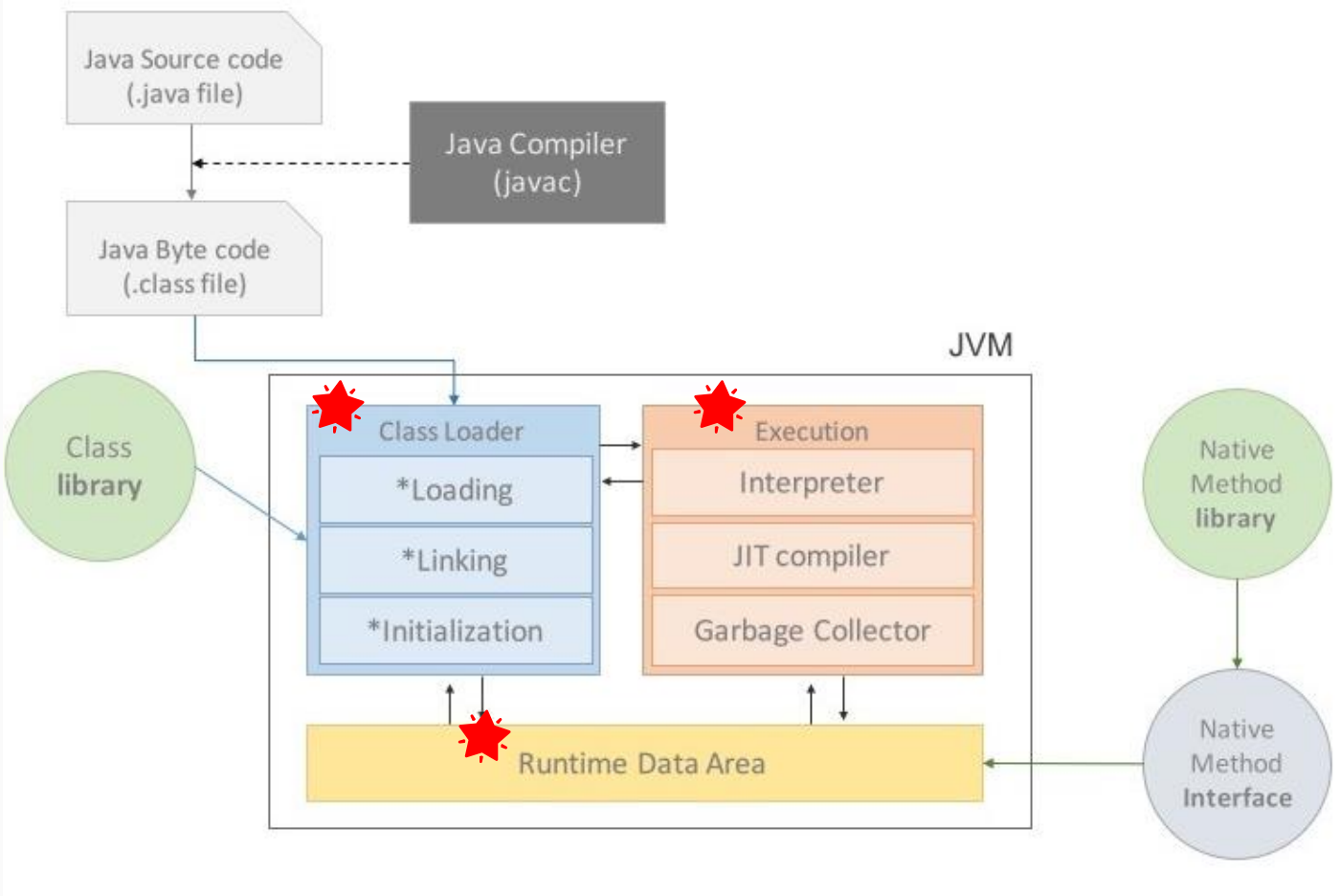
자바 바이트코드는 JRE위에

JRE = 자바 API + JV

실행하는 가상머신

JVM의 구성

1. Runtime Data Area
2. Execution Engine
3. 클래스 로더



클래스 로더

자바 .class 파일을 JVM으로 로딩

Execution Engine

자바 바이트 코드를 실행하는 Runtime Module

자바 인터프리터, JIT 인터프리터 두 가지를 혼합하여 사용

Runtime Data Area

JVM의 메모리 영역

자바 애플리케이션을 실행할 때 사용되는 데이터를 적재하는 영역

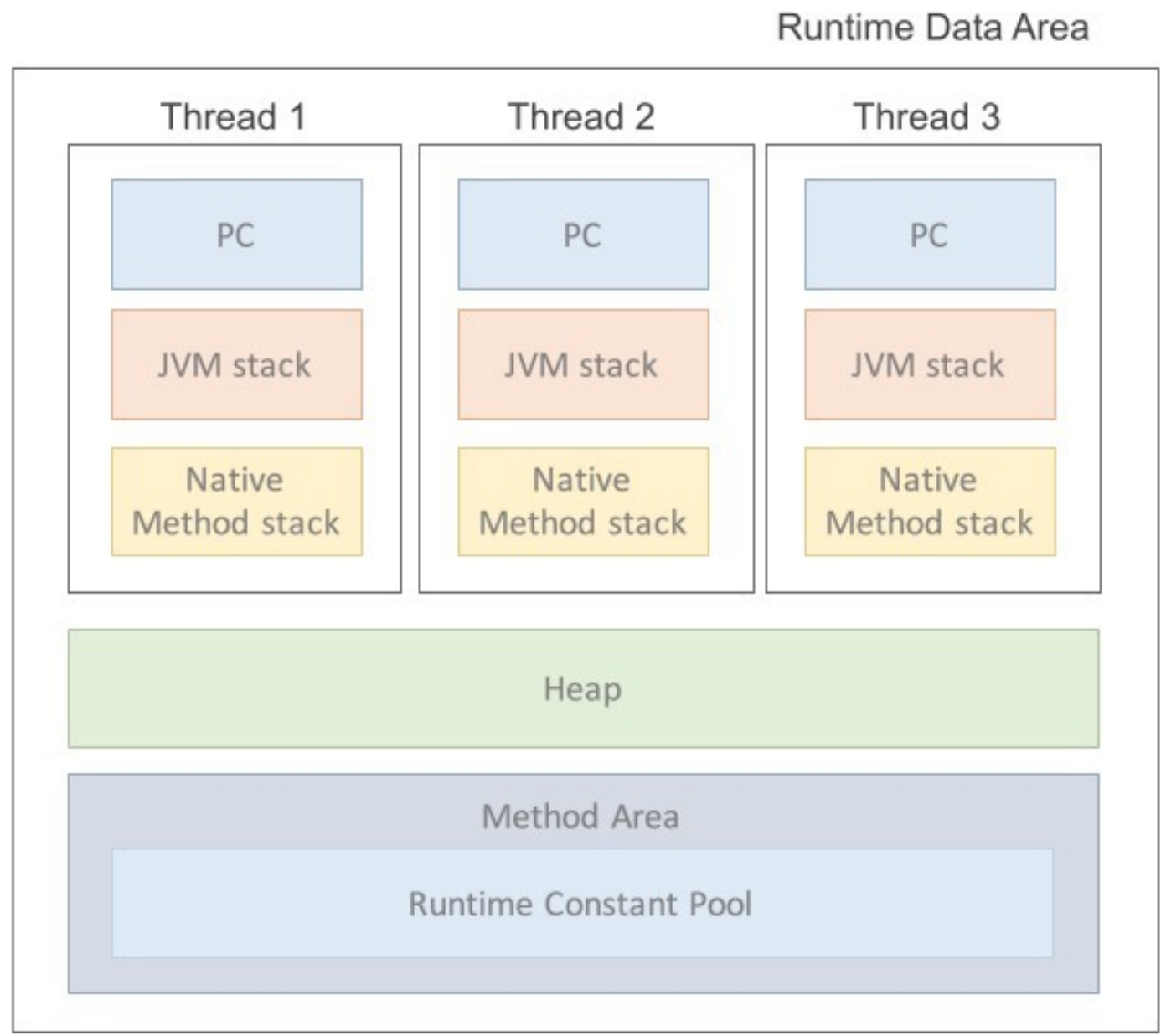
해석된 자바 바이트 코드가 이곳에 배치됨

Runtime

JVM의 메모리 영역

자바 애플리케이션

해석된 자바 바이트



Method Area

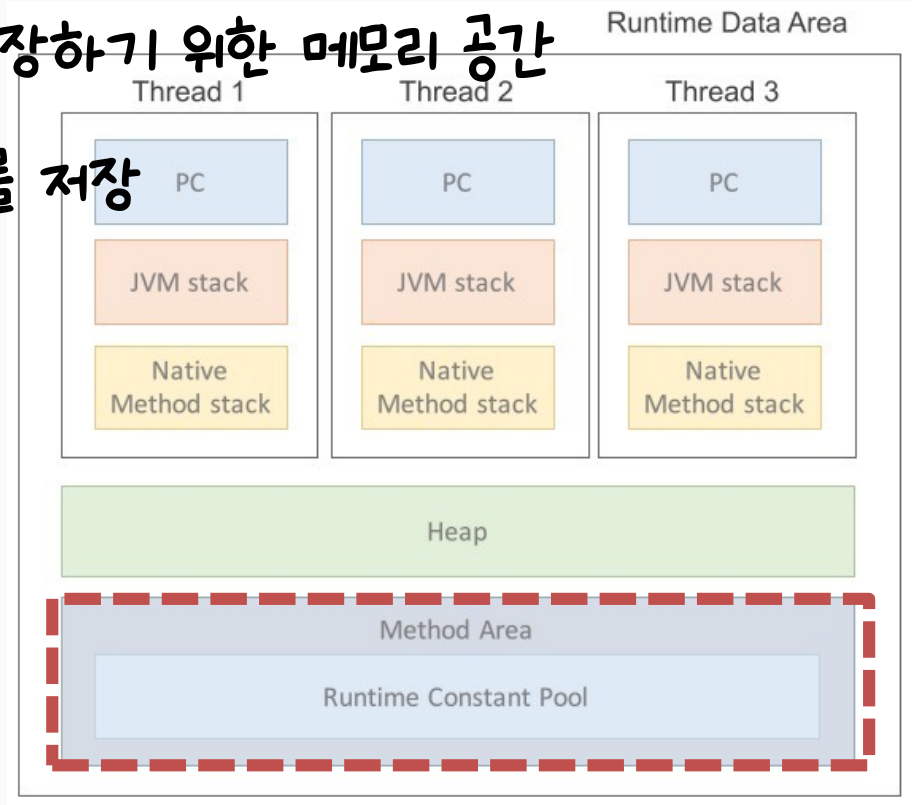
Native Method Stack

Method Area = static area = class area

모든 스레드가 공유하는 영역

클래스 정보를 처음 메모리 공간에 올릴 때 초기화되는 대상을 저장하기 위한 메모리 공간

클래스, 인터페이스, 메서드, 필드, static 변수 등의 바이트 코드를 저장



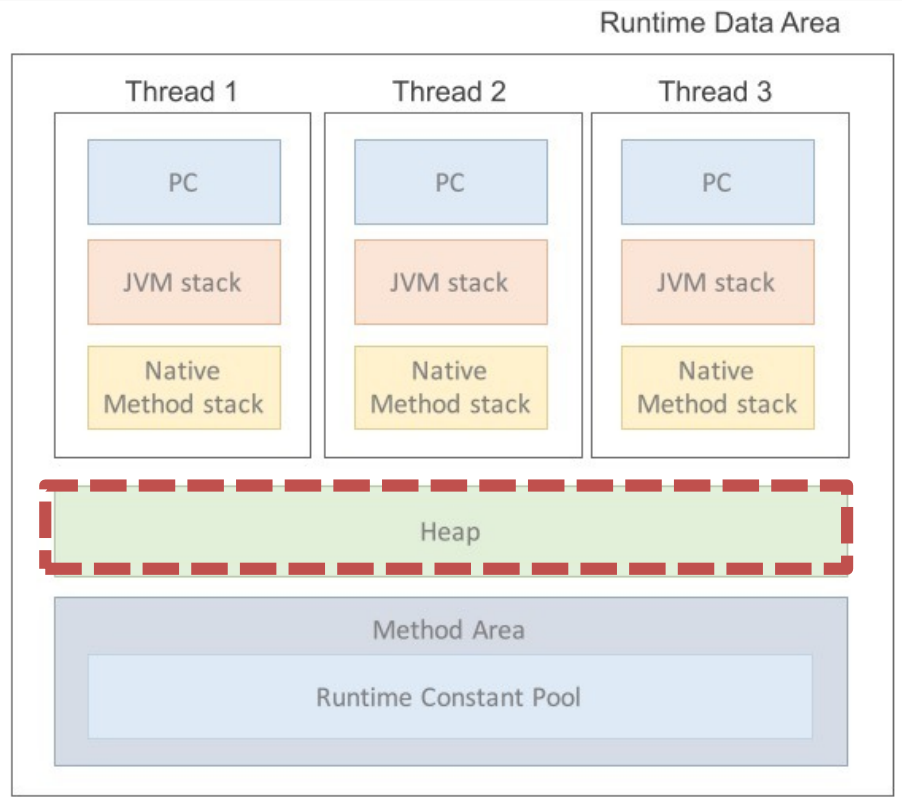
Heap Area

객체를 저장하는 가상 메모리 공간

모든 스레드가 공유, new 키워드로 생성된 객체와 배열을 저장

메서드 영역에 로드된 클래스만 생성 가능

GC가 참조되지 않는 메모리를 확인하고 제거

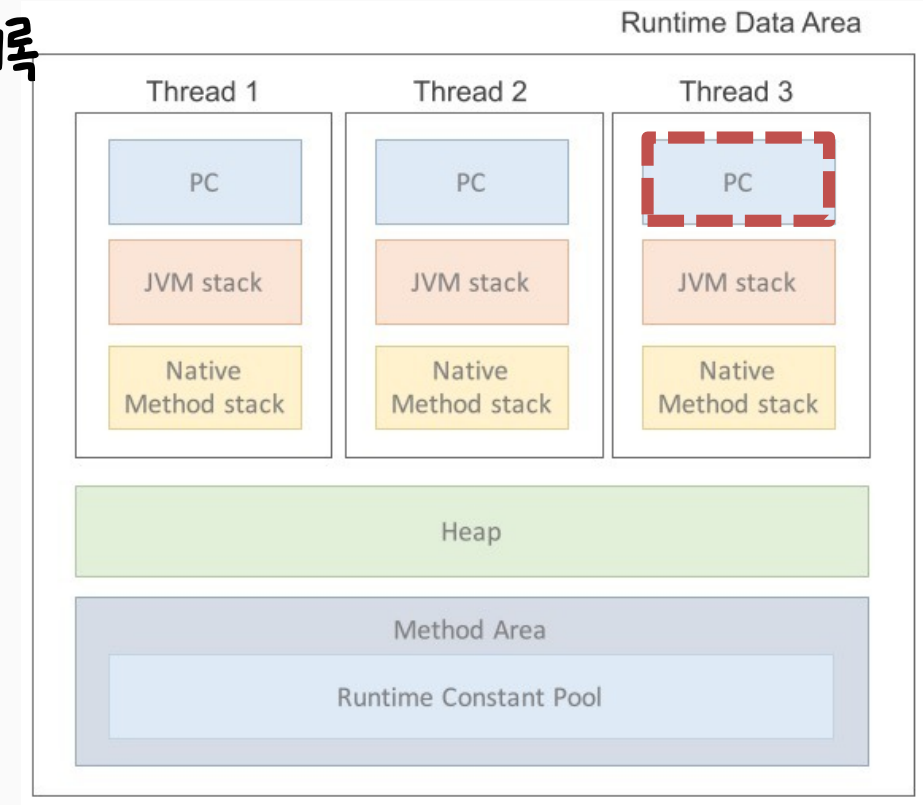


PC Register

스레드가 시작될 때마다 생성되는 공간으로 스레드마다 각각 하나씩 존재

스레드가 어떤 부분을 무슨 명령으로 실행해야 할 지에 대한 기록

현재 수행중인 JVM 명령의 주소를 가짐

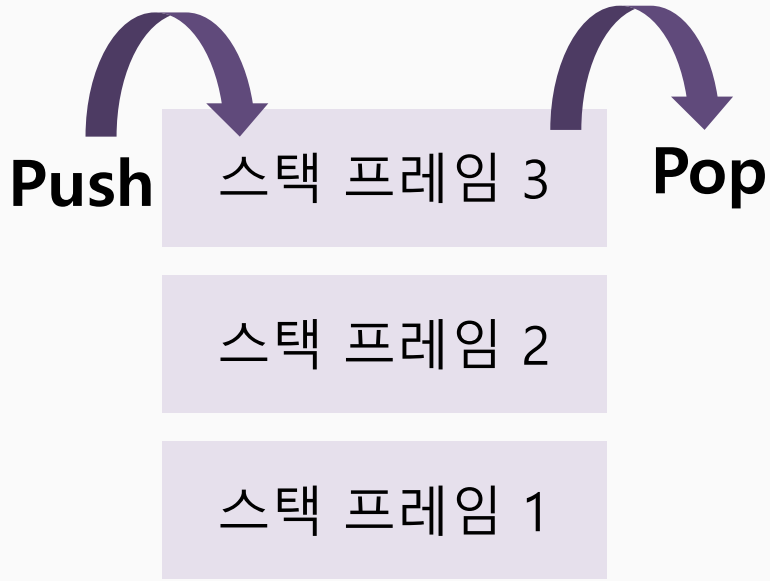


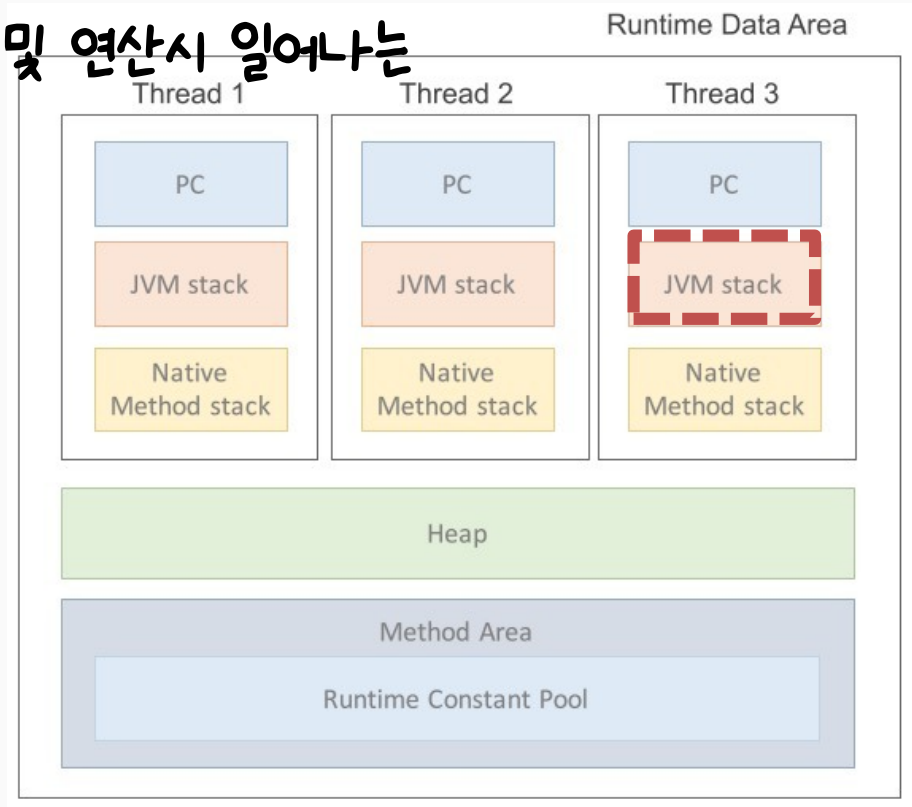
Stack Area

메서드 호출 시마다 각각의 스택 프레임이 생성

메서드 안에서 사용되는 값, 메서드의 매개/지역변수, 리턴 값 및 연산시 일어나는

값을 임시로 저장





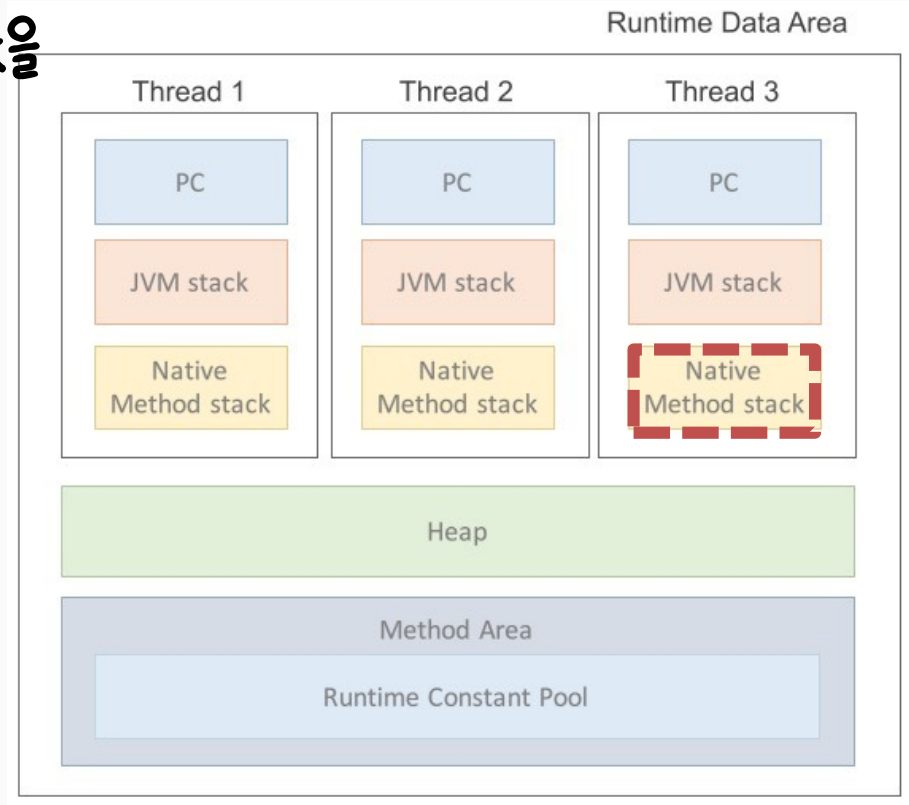
Native method stack

자바 외 언어로 작성된 네이티브 코드를 위한 메모리 영역

스레드가 native method stack을 호출하면 java stack을

벗어나게 되고, 이 때 native method stack이 사용됨

C, C++과 같은 언어로 구현된 메소드이다.



CS study 17주차

다음 시간에..