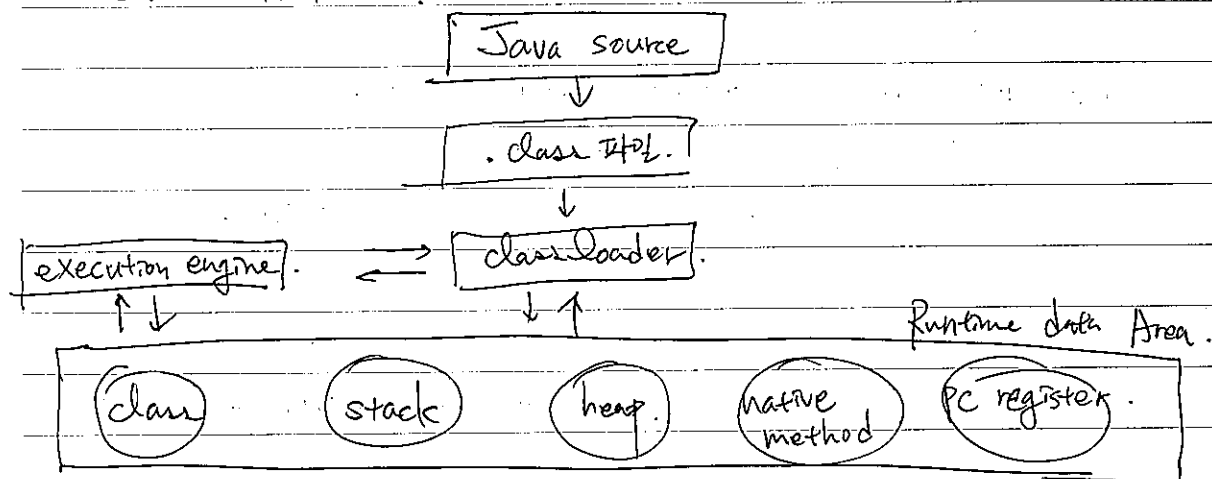


< Java 1.4 → 1.5 >

1. Java Generics 사용 가능.
2. Auto boxing / Unboxing 가능.
Primitive 타입 ↔ Wrapper 타입 번거롭지 않게 value를 넣고 빼는 것이 가능.
3. for each 문 향상
4. StringBuilder 추가.

< Java 1.7 → 1.8 >

1. 람다식.
 2. 메타데이터 지원, 보안. Permanent Generation
 3. Heap 영역에서 Perm Generation 제거
- * JVM 메모리 구조



- class 영역 : field, method, type, constant, static 변수의 정보
- stack 영역 : 지역변수, 매개변수, return 값, return 주소
- heap 영역 : new 연산자로 생성된 객체나 배열을 저장함.
Garbage collector에 의해 관리되는 영역.

→ Permanent 영역 : دينامي하게 코드와 클래스의 대한 메타데이터 및 static 변수나 상수들이 저장되는 공간.

1.8 기준으로 사라지게 되었는데 메타데이터는 Native 메모리를 이용하여 된 (metaspace) 개발자들이 이 영역에 대한 학습 신경 X 여드림
OS에 의해 관리

自然別曲 ^{manip.}

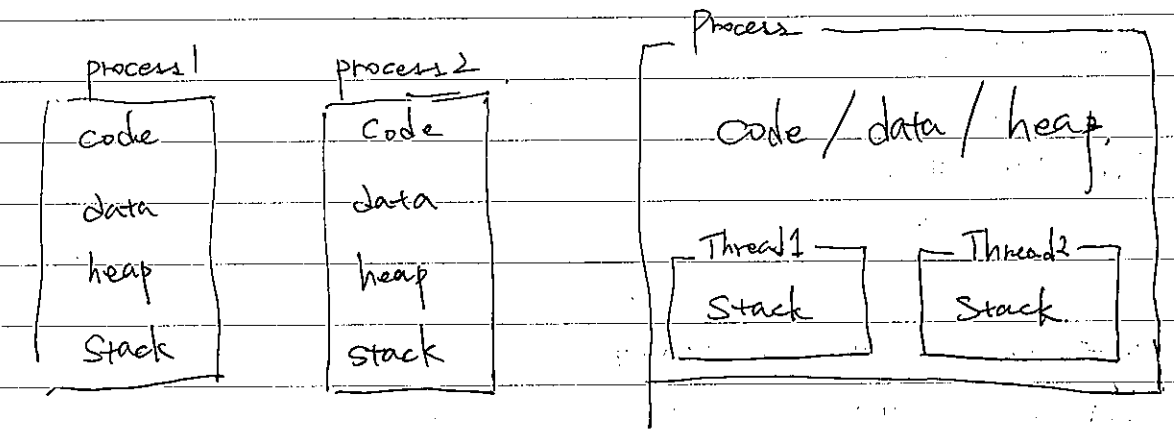
< Process vs Thread >

* Process

컴퓨터에서 연속적으로 실행되고 있는 프로그램.

* Thread

• 프로세스 내에서 실행되는 여러 실행의 흐름.



* 메모리 영역별 용도.

- Stack 영역 : 지역변수, 매개변수, 리턴값, 리턴 주소
런타임에 그 크기를 변경할 수 없다.
- Heap 영역 : 동적 데이터 영역, new, malloc() 함수 등 이용하여 할당되는 영역. Runtime.
- Data 영역 : 정적변수, 정적 변수, 정적 배열, 구조체 등이 저장됨.
- Code 영역 : 프로세스가 실행할 코드와 매진 상수가 기계어 형태로 저장됨.