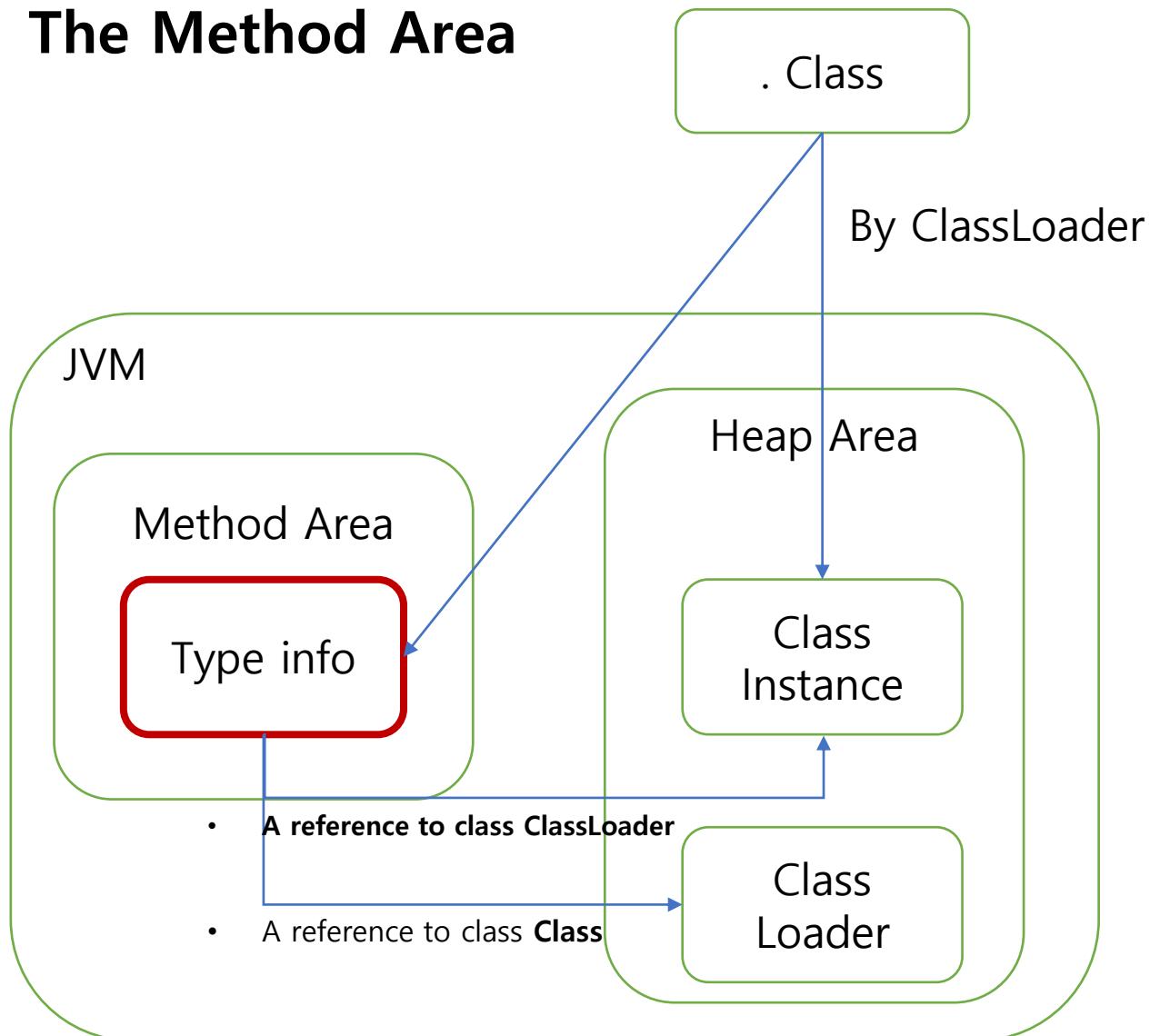


The Method Area



The virtual machine will use the **type information** stored in the method area as it executes the application it is hosting

Type info

- The fully qualified name of the type
- The fully qualified name of the type's direct superclass
- Whether or not the type is a class or an interface
- public, abstract, final
- An ordered list of the fully qualified names of any direct superinterfaces
- The **constant pool** for the type
- **Field** information
- **Method** information
- All class (**static**) **variables** declared in the type, **except constants**
- **A reference to class ClassLoader**
- **A reference to class Class**

An Example of Method Area Use

```
class Lava {  
    private int speed = 5;  
  
    void flow() {}  
}  
  
class Volcano {  
    public static void main(String[] args) {  
        Lava lava = new Lava();  
        lava.flow();  
    }  
}
```

Main() method 호출 단계

- ① give the name "Volcano" to a Java Virtual Machine
- ② Application Class Loader 동작
Application Class Loader 위임 → Extension Class Loader 위임 → BootStrap Class Loader 위임
- ③ BootStrap Class Loader에게 위임되었을 때, 기본적으로 필요한 Class Load!
- ④ Volcano는 Application Class Loader가 Load
- ⑤ Method에 Volcano 정보 까지 존재하면, JVM **main()** method 호출

⚠ main()이 실행 되었음에도 Lava Class는 Load되지 않는다.

An Example of Method Area Use

```
class Lava {  
    private int speed = 5;  
  
    void flow() {}  
}  
  
class Volcano {  
    public static void main(String[] args) {  
        Lava lava = new Lava();  
        lava.flow(); ① 이 라인 수행  
    }  
}
```

- ③ Method에 Load된 Lava Class가 없는 경우

Class Loader 즉, Volcano를 Load한 Application Class Loader가 “Lava”라는 문자를 통해 Lava Class를 Load해온다.

- ④ Symbolic Reference Resolve

5 Class = Lava 주소 값 :: Method 영역에 올라와 있다.

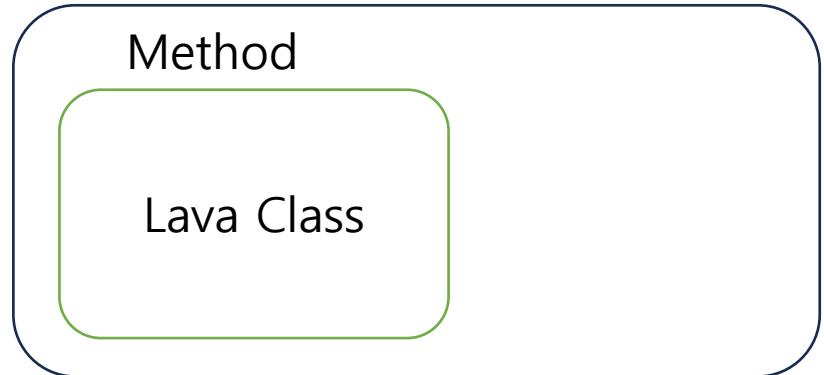
5 Class = Lava

- ② Constant Pool 확인 && method Area에 Lava Class가 Load 되어있는지 확인



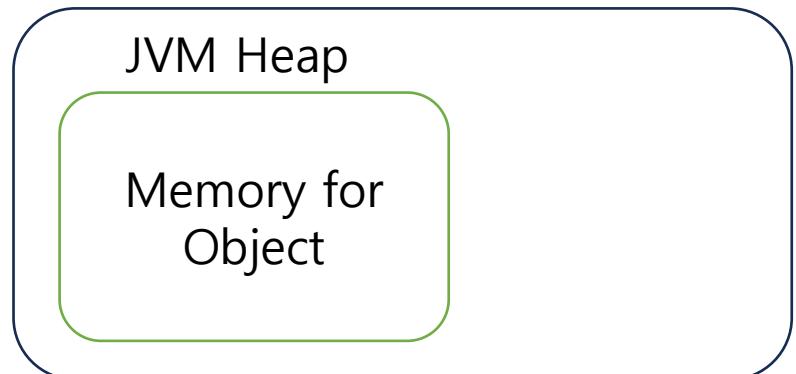
An Example of Method Area Use

JVM



- ① virtual machine consults the information stored in the method area to find out how much heap space is required by a Lava object.

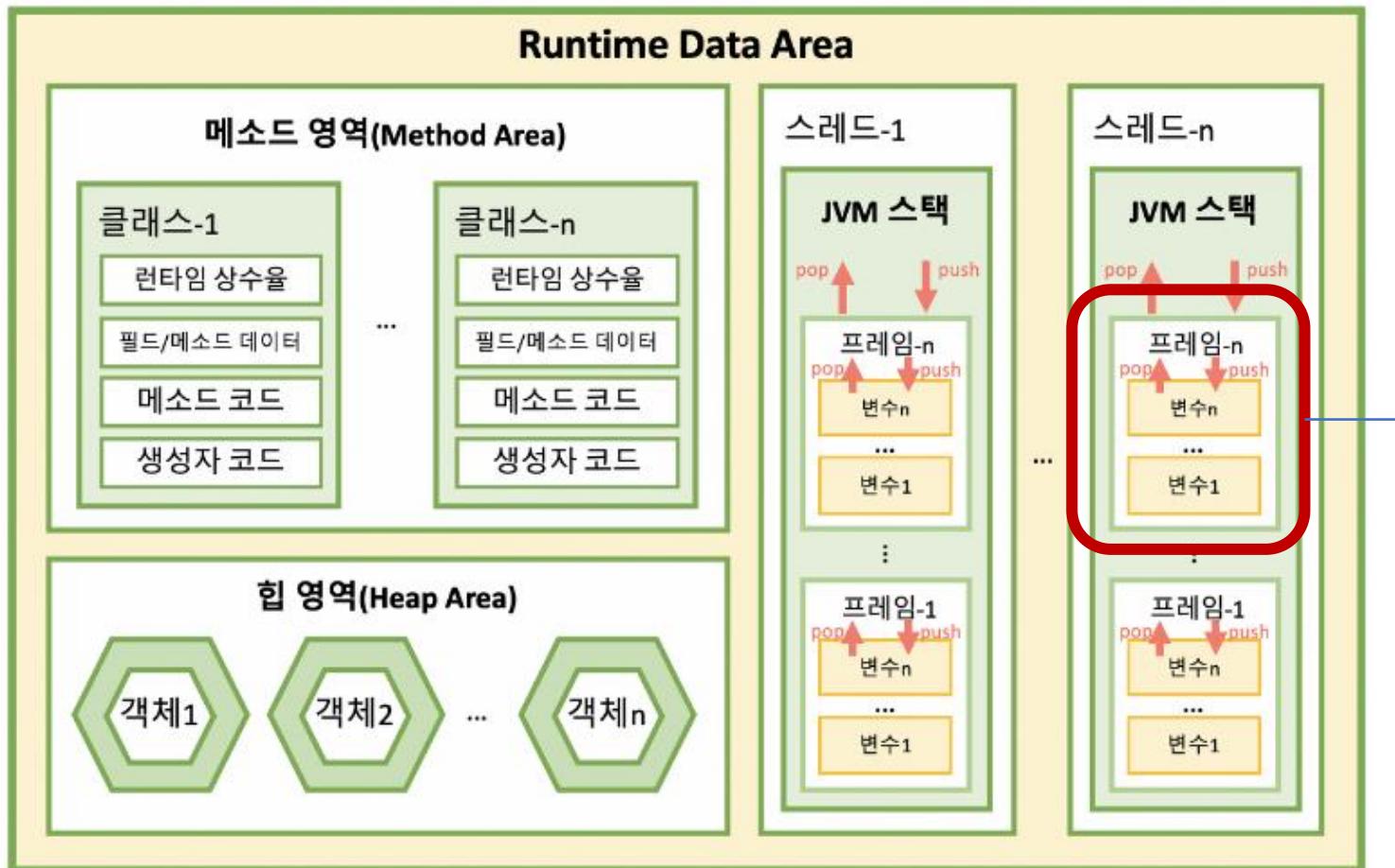
JVM



- ② the virtual machine is ready to actually allocate memory for a new Lava object.

Stack Frame in Java Stack

JVM Memory 전체적인 구조



Stack Frame

Operand Stack
:: 계산 작업하는 공간

Local Variable

):: 지역 변수저장 공간
):: instance method의 경우 0번
index에 this에 대한 ref가 들어간다.

Frame Data

data to support constant pool resolution,
normal method return,
and exception dispatch

Frame Data In Stack Frame

Frame Data

