Java Fundamentos

Germán Valencia

OOP

```
Carro c = new Carro();
```

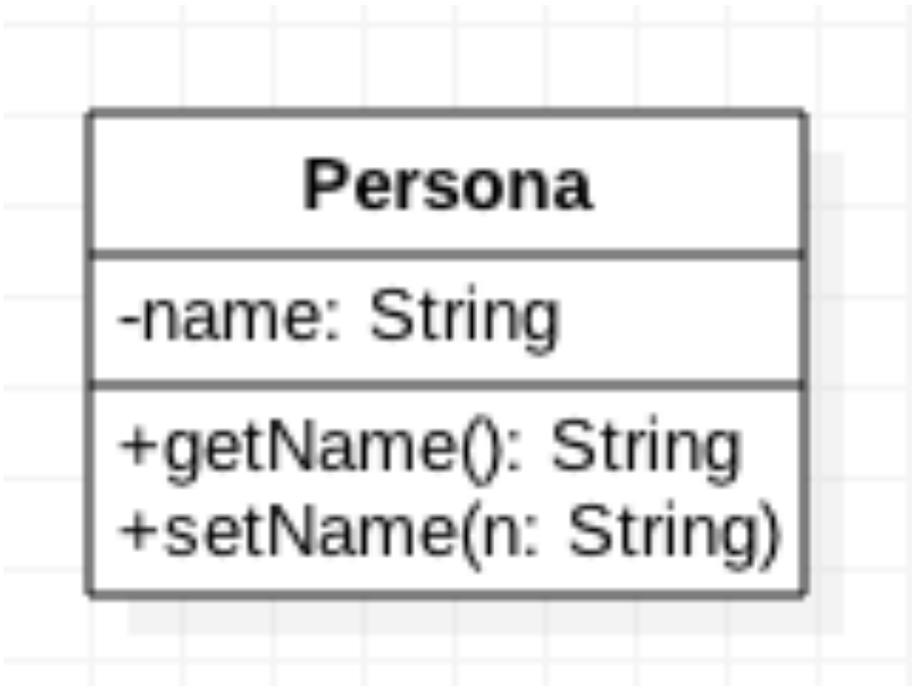
Clase: Template que define attributos y métodos

Carro	
Attributes (state)	color, motor, puerta, llanta, asiento, gasolina, luz, marca, gearbox
Methods (Behavior)	prender, acelerar, frenar, girarCabrilla, abrirPuerta, apagar.

Objeto: Instancia real de una clase

Principios OOP

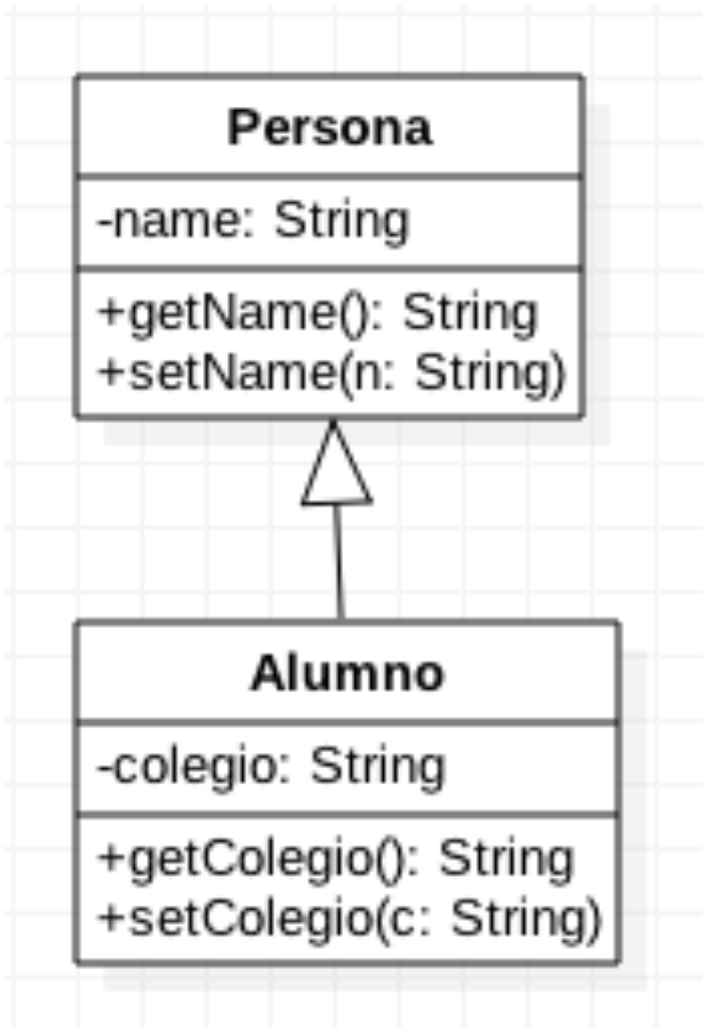
- Encapsulamiento: Los atributos solo se pueden acceder a través de los métodos.




```
private String name;
public String getName();
public void setName(String n);
```

Principios OOP

- Herencia: Relación que permite compartir comportamiento y atributos de forma jerárquica.

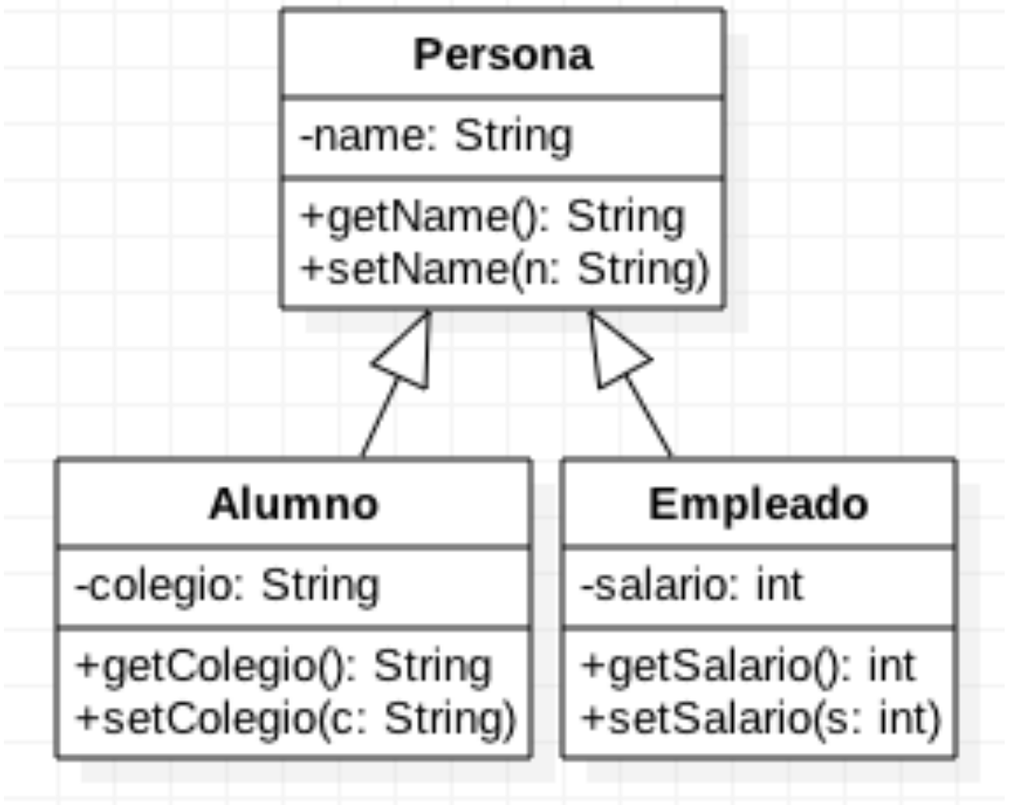



```
class Alumno extends Persona
```

Principios OOP

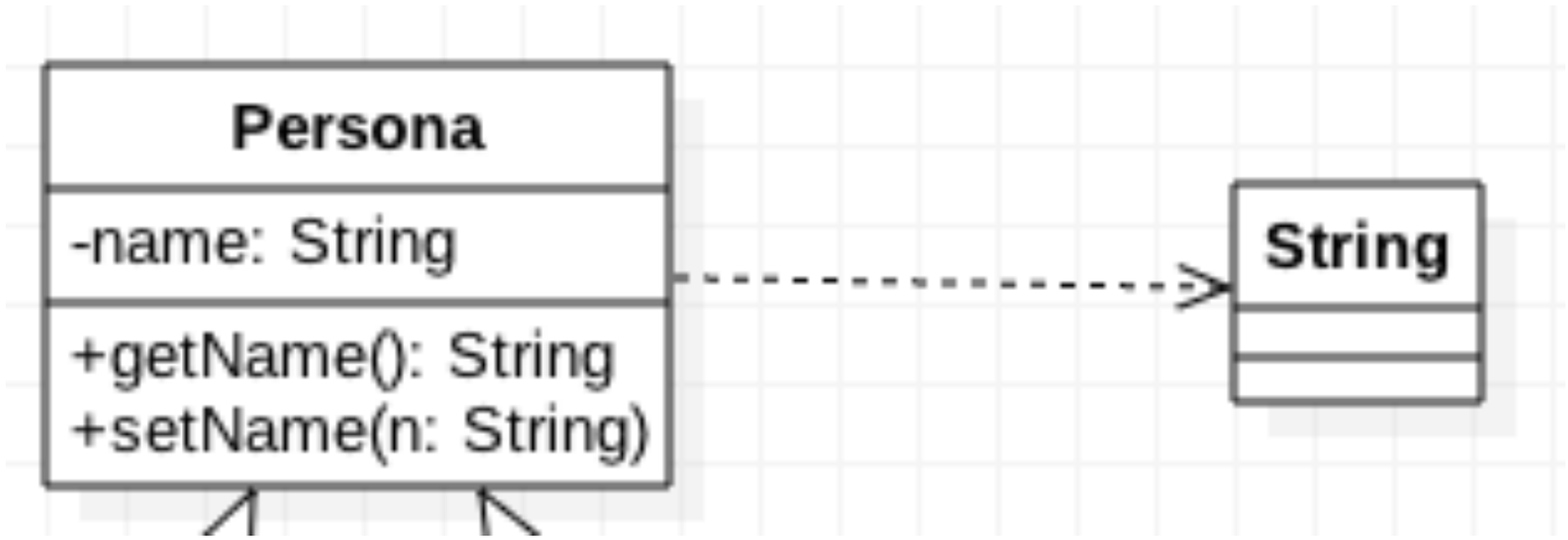
- Polimorfismo: Capacidad que tiene una clase padre de obtener cualquier instancia de sus clases hijas.

```
Persona p = new Persona();
Persona pa = new Alumno();
Persona pe = new Empleado();
Alumno ae = new Empleado();
```



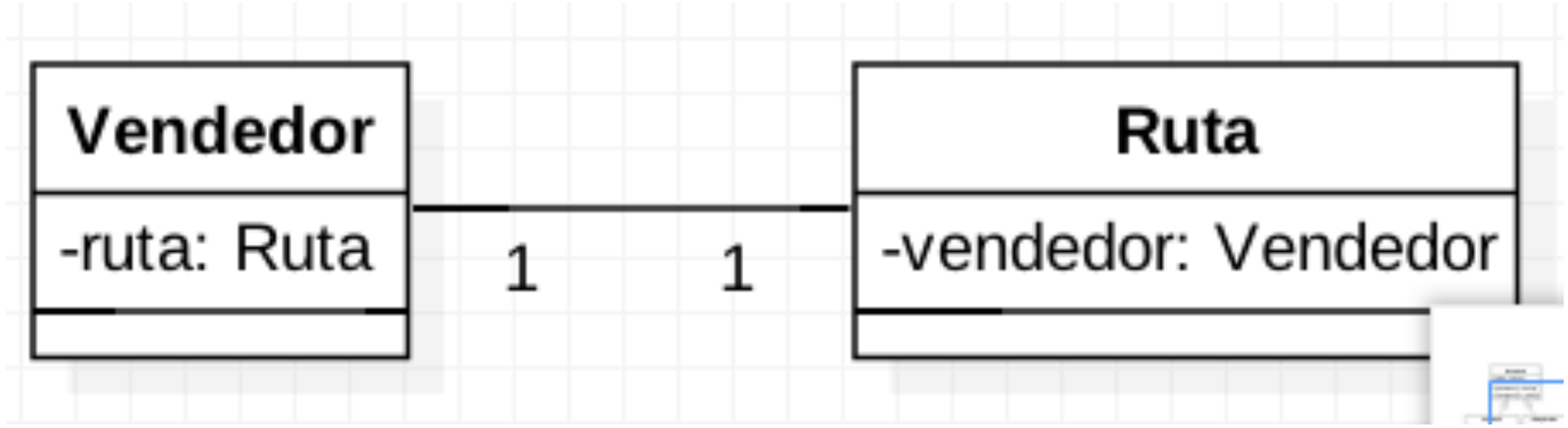
Relaciones UML

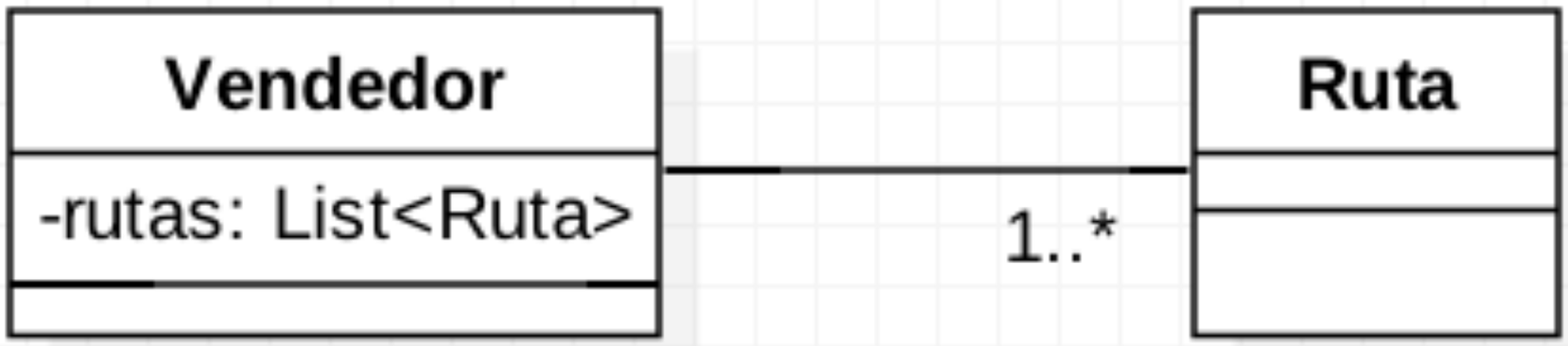
- Dependencia: Relación débil que denota simple uso



UML

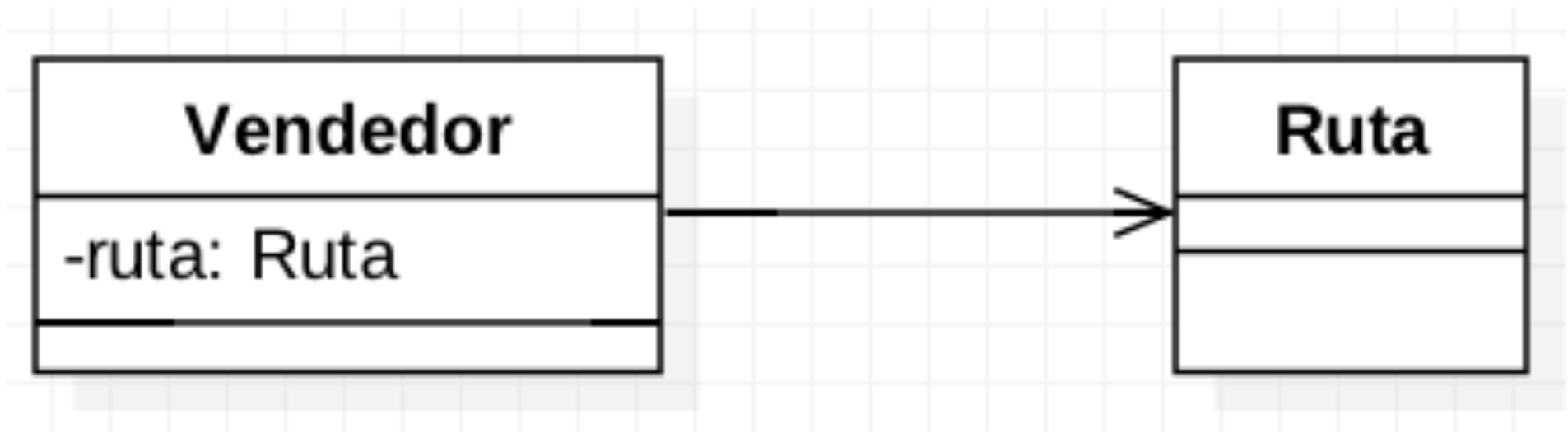
- Asociación: Relación de dependencia fuerte que además define multiplicidad.





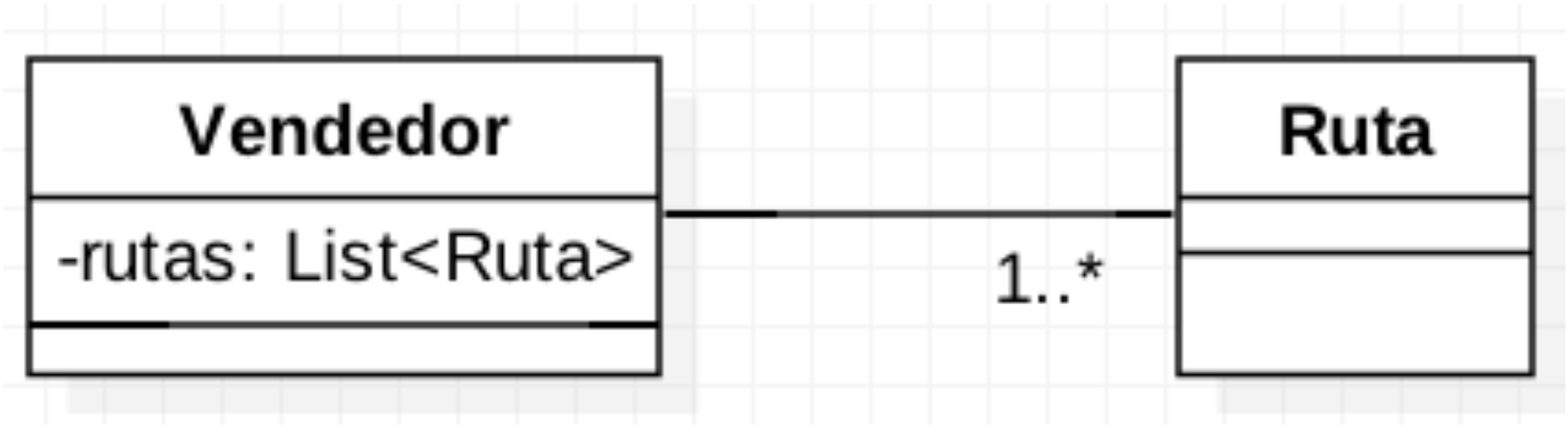
UML

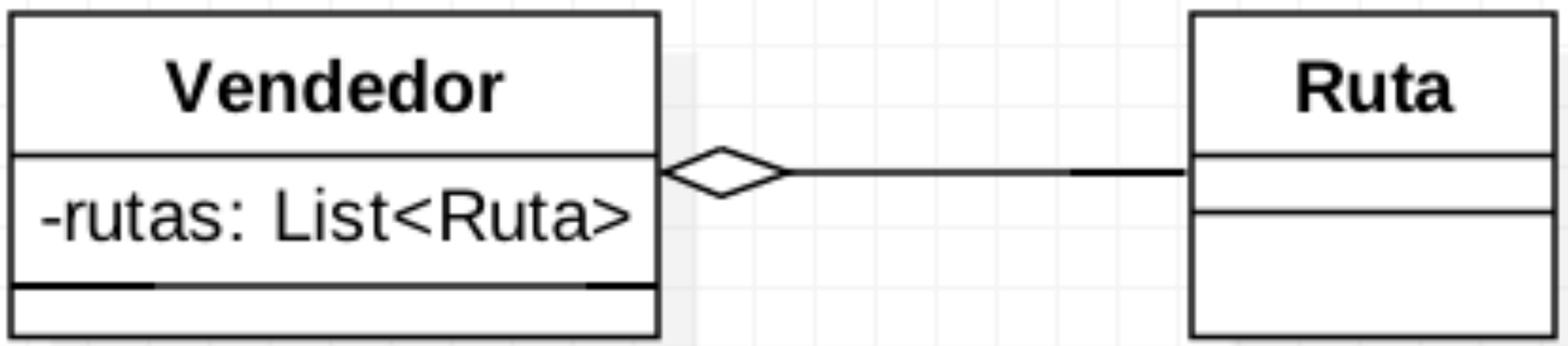
- Asociación Directa: Relación explicita entre el objeto contenido y el que lo contiene



UML

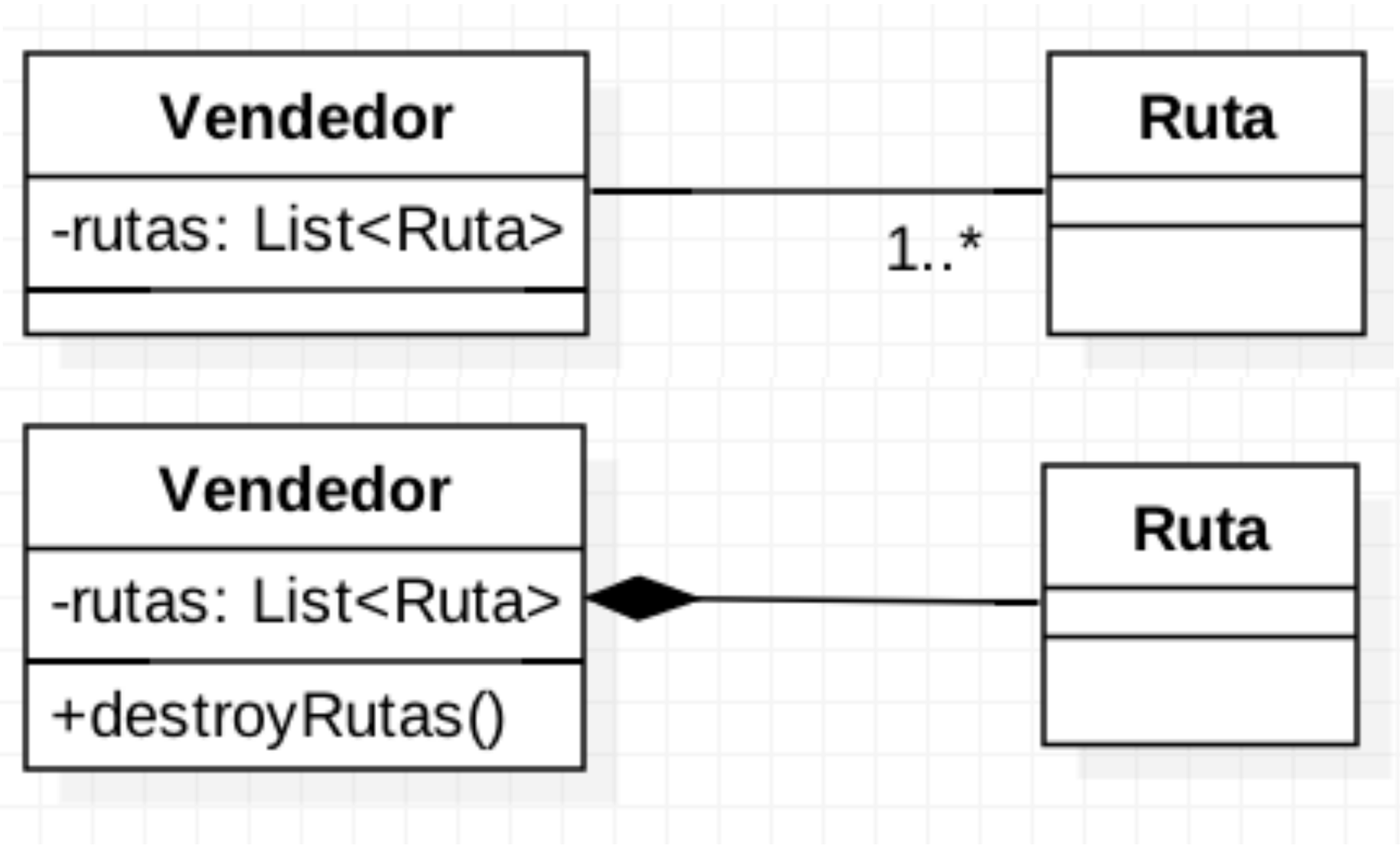
- Agregación: Asociación de multiplicidad 0..* o 1..*. El ciclo de vida de los objetos contenidos ***no depende*** del padre.





UML

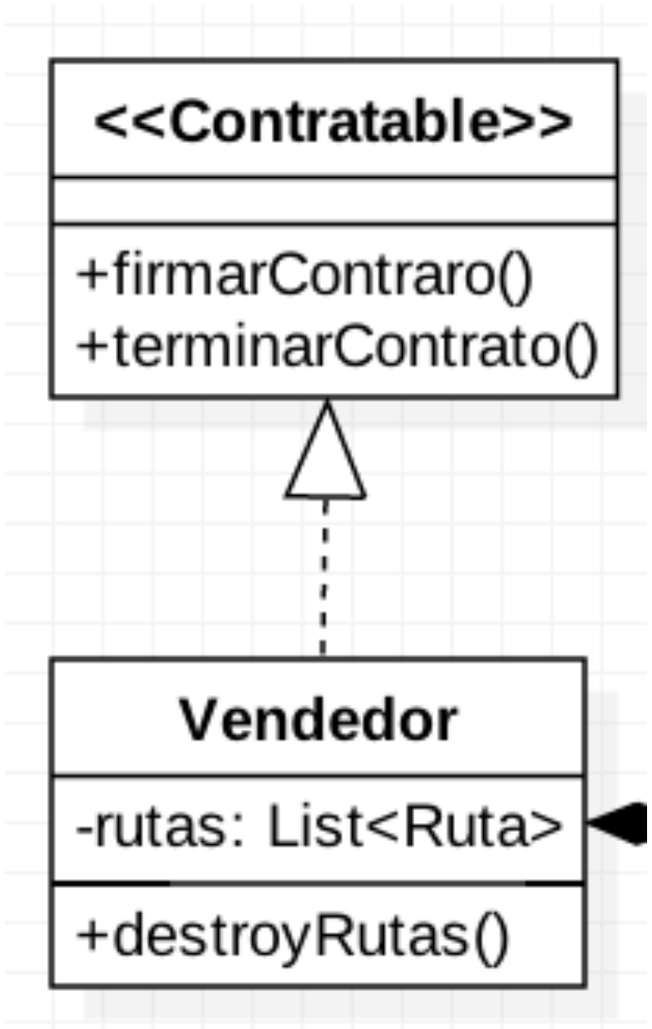
- Composición: Asociación de multiplicidad 0..* o 1..*. El ciclo de vida de los objetos contenidos ***depende*** del padre.



UML

- Realización: Implementación de una interface.

```
class Vendedor implements Contratable
```



Abstract Class & Interface

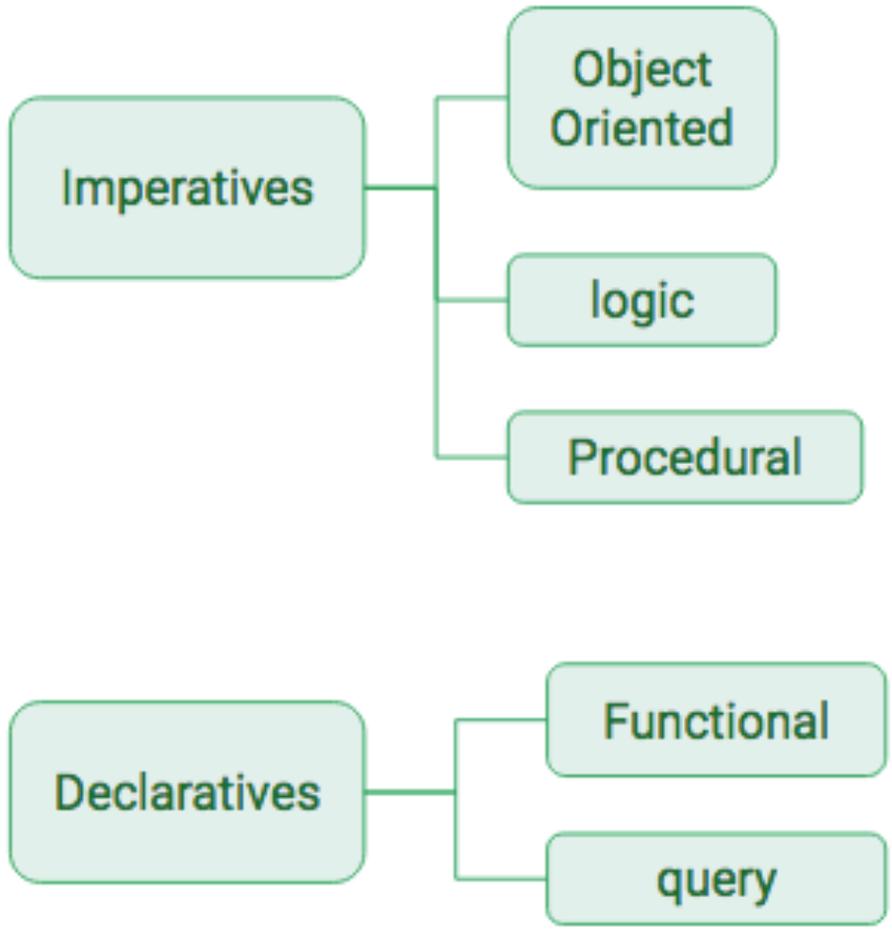
- Tanto las clases abstractas como las interfaces ofrecen el nivel de abastracción más alto en POO.

- Ninguna de las 2 pueden ser instanceadas, por lo tanto solo podrán mantener referencias de sus clases hijas, y en el caso de las interfaces cualquier clase que la implemente.

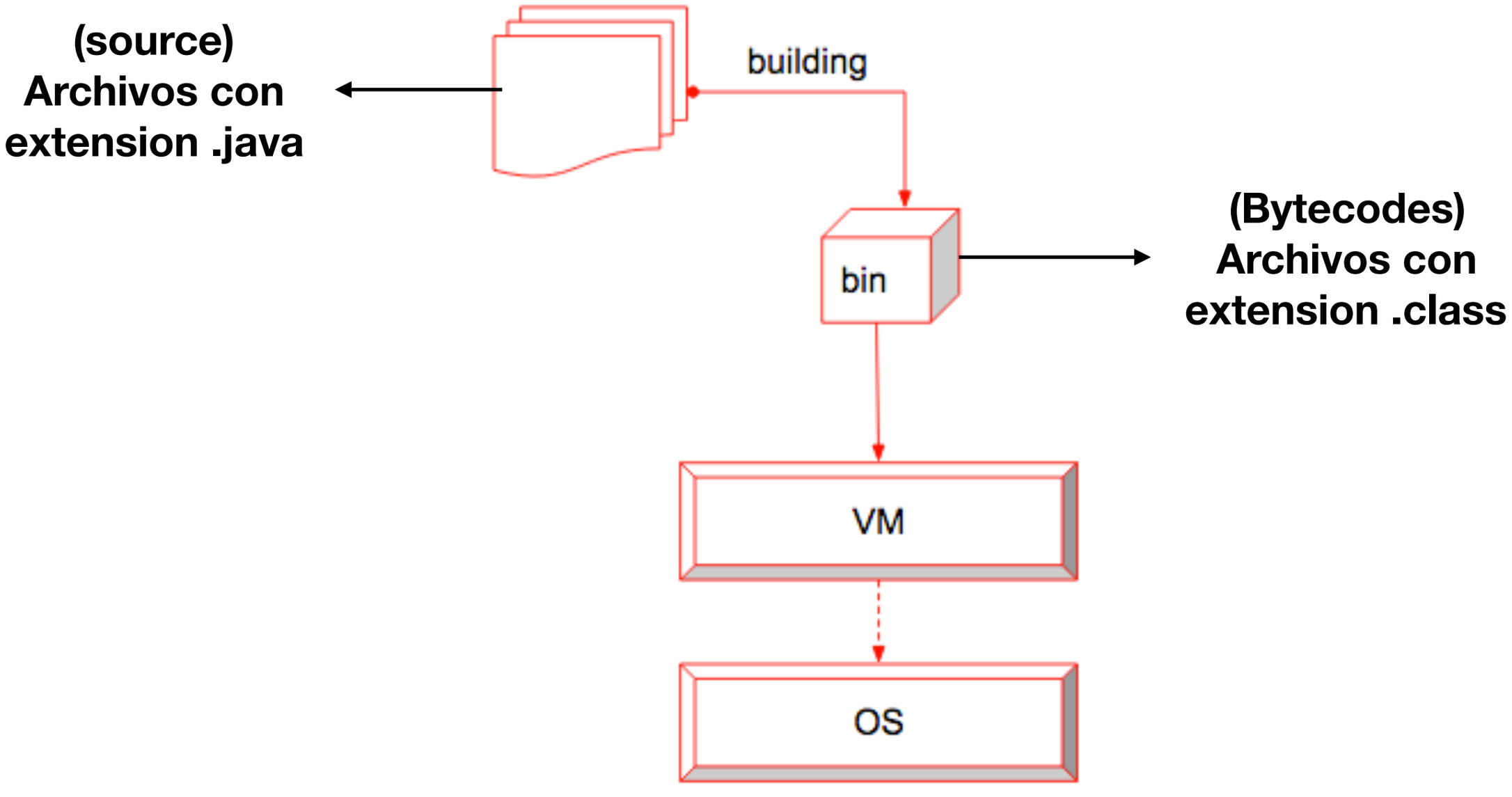
```
abstract class Employee
```

```
interface Hireable
```

JVM Overview



JVM Overview



Parámetros por Valor y Referencia

- Por valor (tipos primitivos + clase String): La variable original y el parámetro de la funcion apuntan a 2 direcciones de memoria distintas

- Por referencia (cualquier otra clase): El parámetro de la función es una copia de la referencia a la que apunta la variable original.

Estructuras de datos

- Arreglos: Listas de elementos con longitud finita.
- Una Colección representa una lista iterable de elementos, y de crecimiento dinámico.
- Un Mapa representa una lista de elementos ***llave - valor*** (key-value), y de crecimiento dinámico.

Java Modifiers

Modifier	Class	Method	attribute	Local field
public	ok	ok	ok	—
private	—	ok	ok	—
static	ok (inner)	ok	ok	—
final	ok	ok	ok	ok
abstract	ok	ok	—	—

Ordered and Sorted

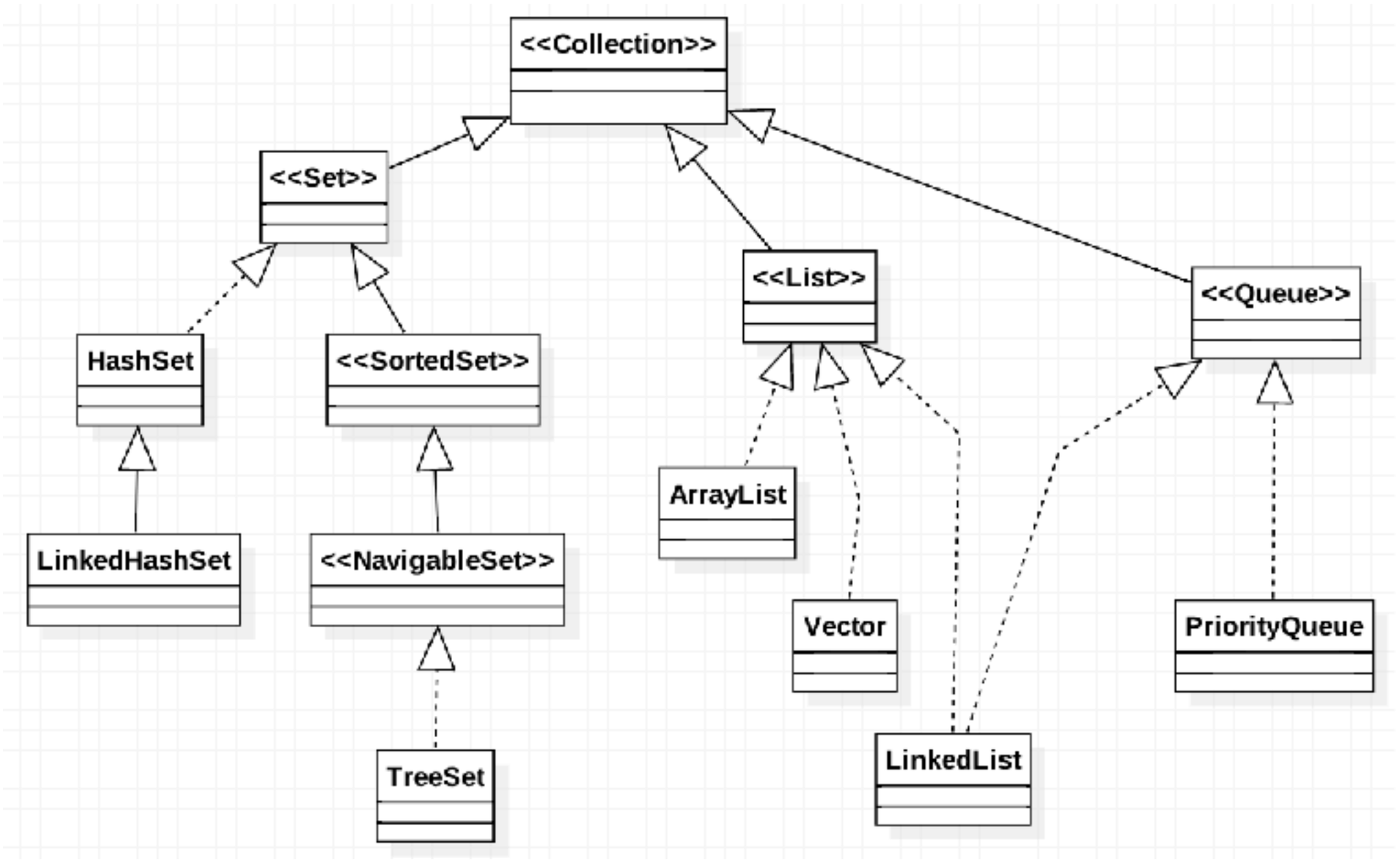
- La colección permite recorrer sus elementos en un orden específico a través de un indice.

index	0	1	2	3	4
value	100	300	200	700	500

- La colección ordenada que además permite organizar los elementos a través de un criterio de ordenamiento.

index	0	1	2	3	4
value	100	200	300	500	700

Collections



Collection flavors

- <<List>>: Representan listas de elementos (duplicados), que se asocian a un índice.

Implementation	Ordered	Sorted	Key points
ArrayList	Yes	No	• Single linked-list • Fast random access
Vector	Yes	No	• Similar to an ArrayList • A vector is Synchronized. • thread-safe affects performance
LinkedList	Yes	No	• Double linked-list • Fast insertion and deletion • Slow to iterate

Collection flavors

- <<Set>>: Representan listas de elementos únicos

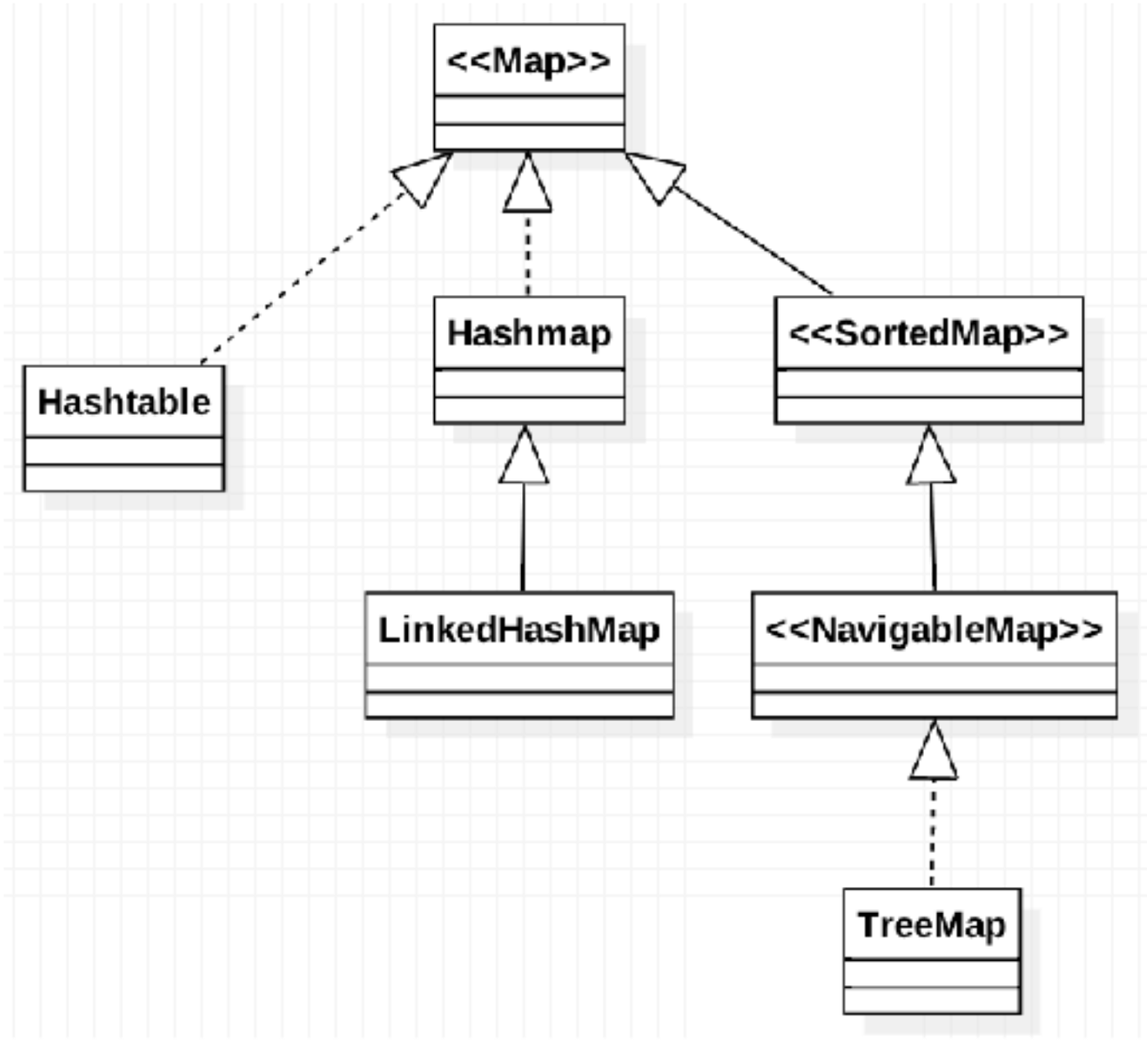
Implementation	Ordered	Sorted	Key points
HashSet	No	No	• No duplicates • You must override equals() • You must override hashCode()
LinkedHashSet	Yes	No	• No duplicates • You must override equals() • You must override hashCode()
TreeSet	Yes	Yes	• Sorted by ascending natural order • Elements must implement <<Comparator>> to set the sorting criteria

Collection flavors

- <<Queue>>: Los elementos se almacenan de acuerdo al orden en que van a ser procesados (FIFO)

Implementation	Ordered	Sorted
ArrayList	Yes, by index	No
Vector	Yes, by index	No
LinkedList	Yes, by index	No

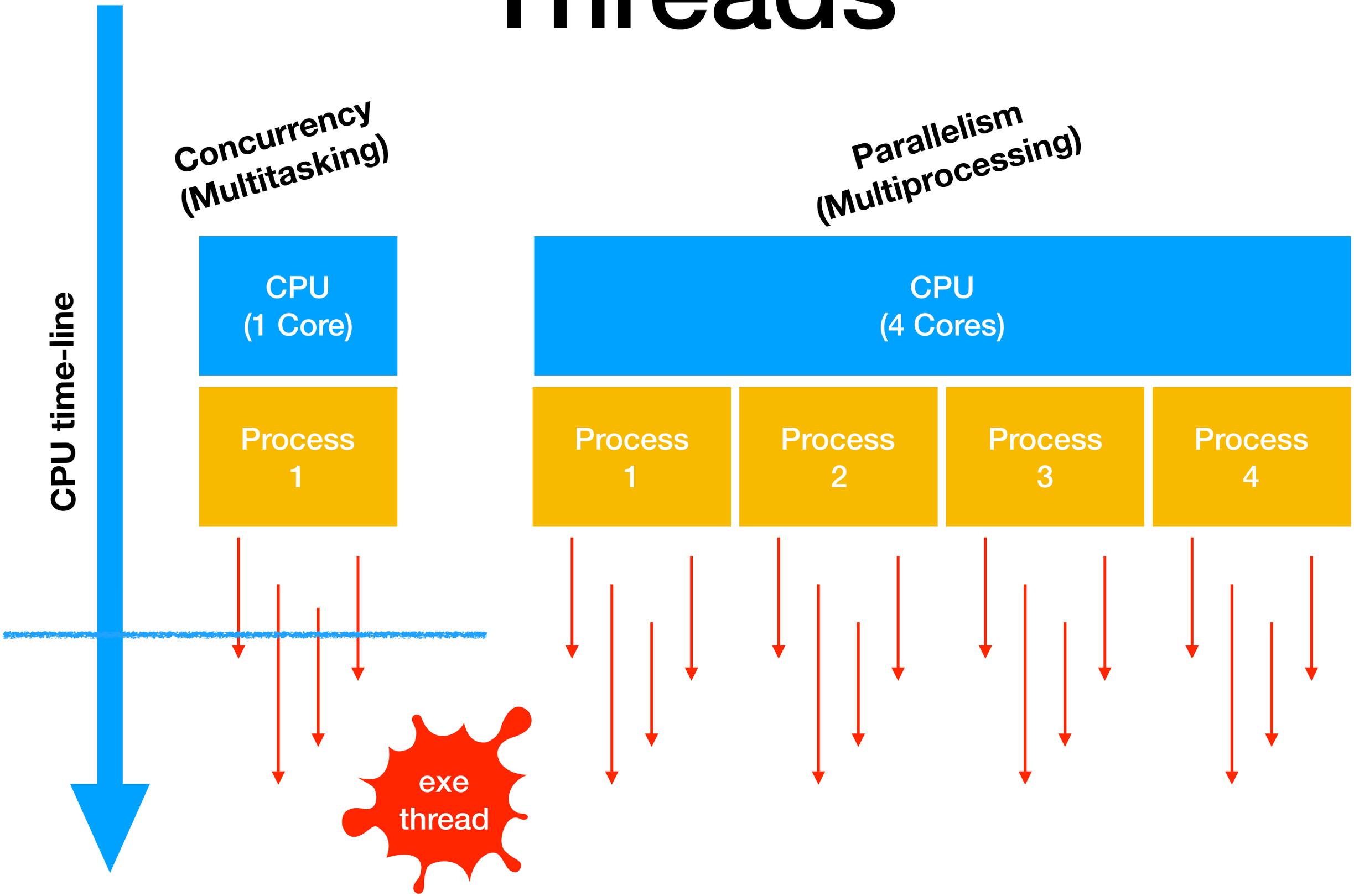
Maps


<<Map>>
Hashtable
Hashmap
<<SortedMap>>
LinkedHashMap
<<NavigableMap>>
TreeMap

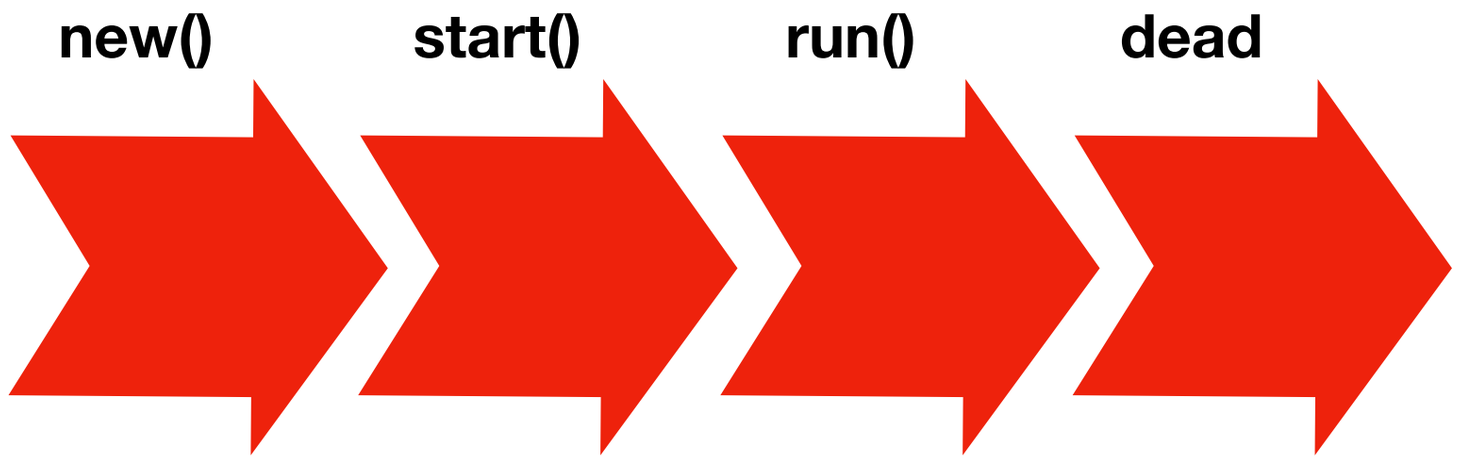
Threads

- Concurrencia: Capacidad que tiene un programa de ejecutar tareas simultáneas, sobre una única unidad de procesamiento (1 Core - Multitasking).
- Paralelismo: Capacidad que tiene un programa de crear procesos que corren sobre diferentes unidades de procesamiento exactamente al mismo tiempo (N cores - multiprocessing).

Threads


CPU time-line
Concurrency
(Multitasking)
CPU
(1 Core)
Process
1
Parallelism
(Multiprocessing)
CPU
(4 Cores)
Process
1
Process
2
Process
3
Process
4
exe
thread

Threads - LifeCycle



Thread Pools in Java

- Executors: Clase utilitaria
- <<Executor>> : Interfaz que permite correr una instancia de un hilo.
- <<ExecutorService>>