

Desarrollo de Software I

Java Collection Framework

Parte I

Ing. Eric Gustavo Coronel Castillo

gcoronelc@gmail.com

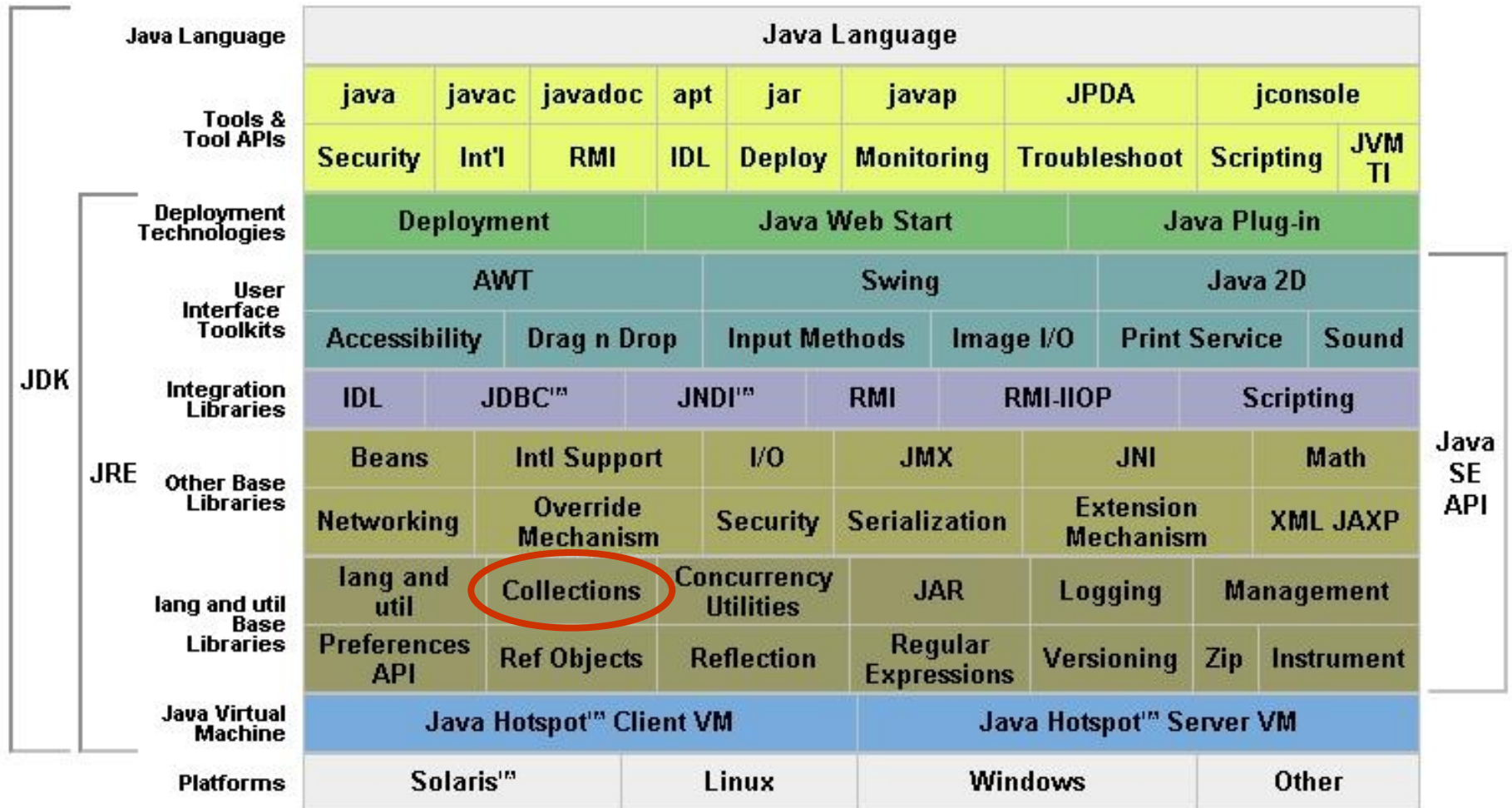
gcoronelc.blogspot.com

Índice

- ❖ Objetivo
- ❖ Introducción
- ❖ Interfaces del Core
- ❖ Implementación del Core

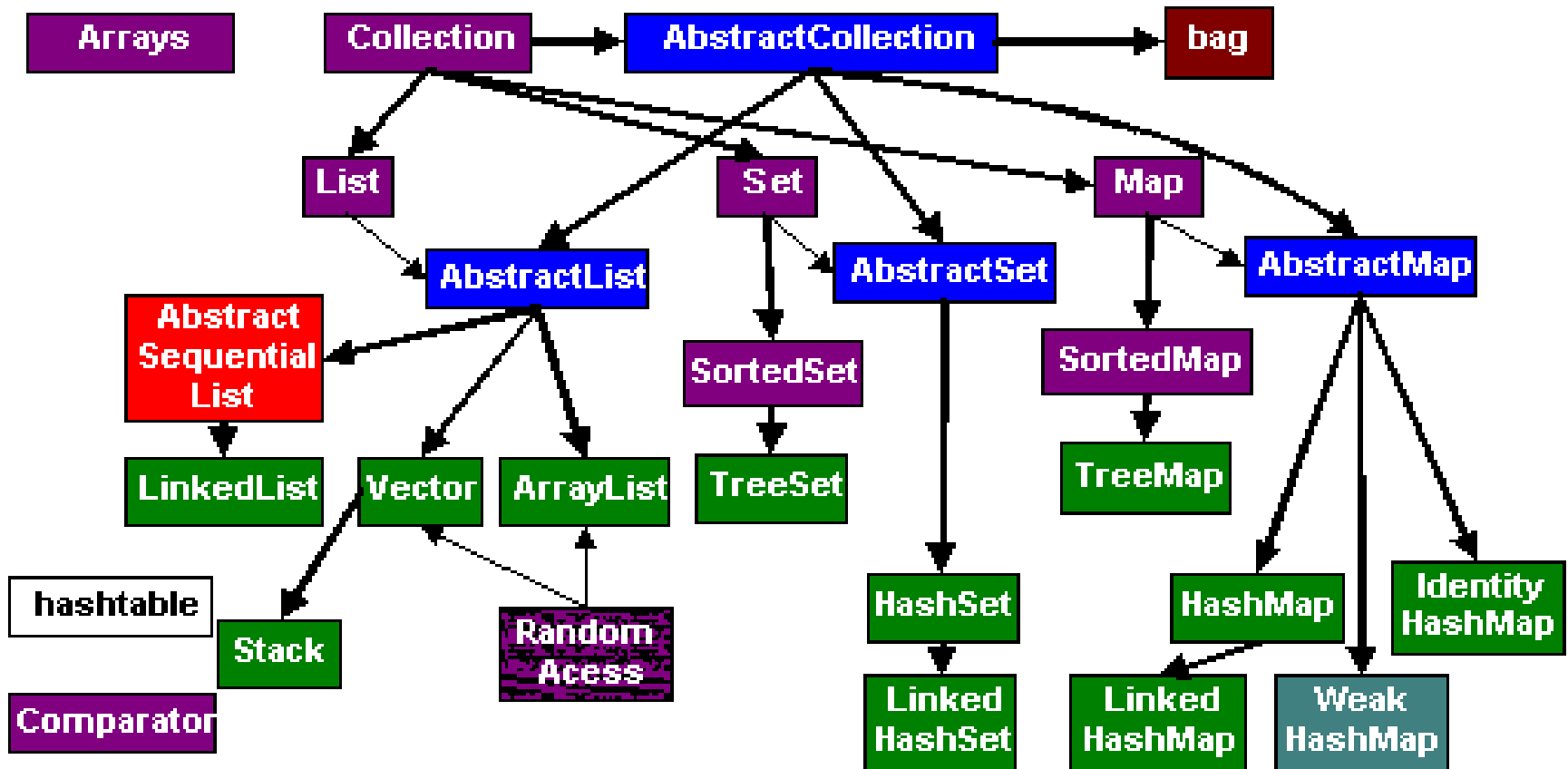
Objetivo

❖ Trabajar colección de datos mediante JCF.



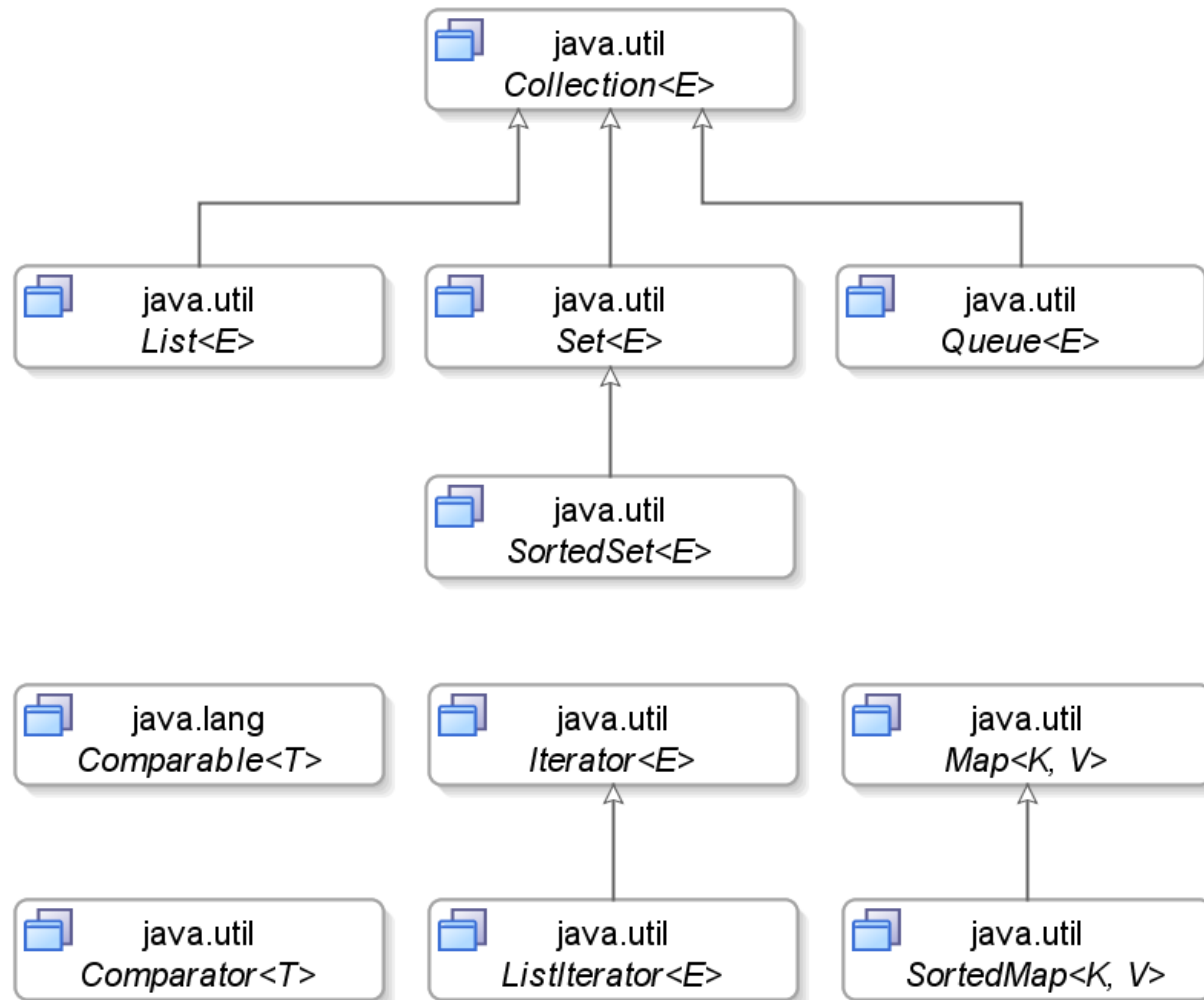
Introducción

- ❖ Un marco de trabajo es un conjunto de interfaces y clases proporcionados para resolver un problema determinado.
- ❖ La intención es utilizar las clases proporcionadas, extenderlas o implementar las interfaces.



Interfaces del Core

Proporcionan métodos para todas las operaciones comunes, y las implementaciones concretas especifican la decisión de las operaciones no permitidas.



Interfaces del Core

❖ Interfaces del Core

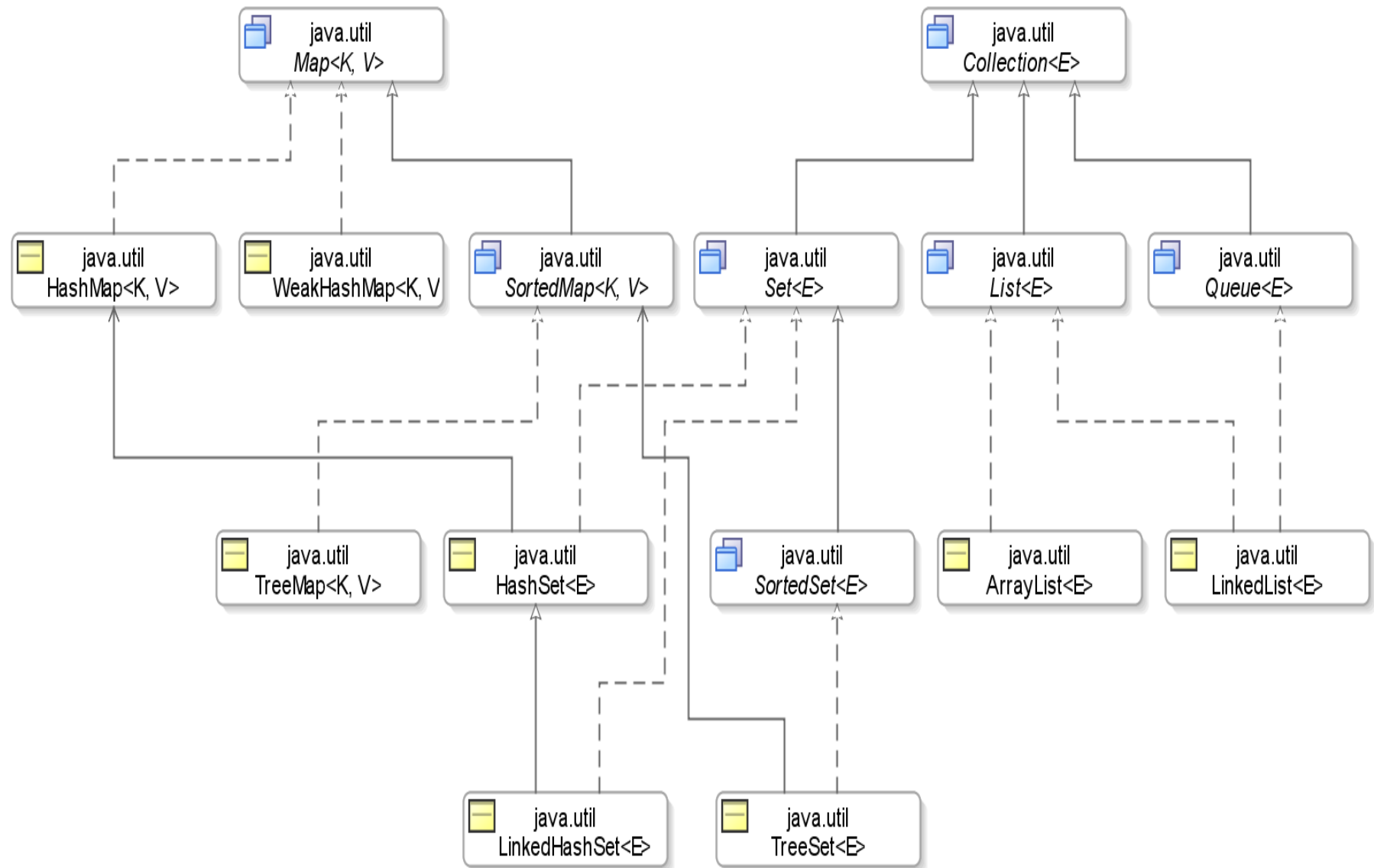
INTERFAZ	DESCRIPCIÓN
Collection	Representa un grupo de objetos sin implementaciones directas, agrupa la funcionalidad general que todas las colecciones ofrecen.
Set	Colección que no puede tener objetos duplicados.
SortedSet	Set que mantiene los elementos ordenados
List	Colección ordenada que puede tener objetos duplicados
Map	Colección que enlaza claves y valores; no puede tener claves duplicadas y cada clave debe tener al menos un valor.
SortedMap	Map que mantiene las claves ordenadas.
Queue	Colección que maneja la prioridad para procesar los elementos

Interfaces del Core

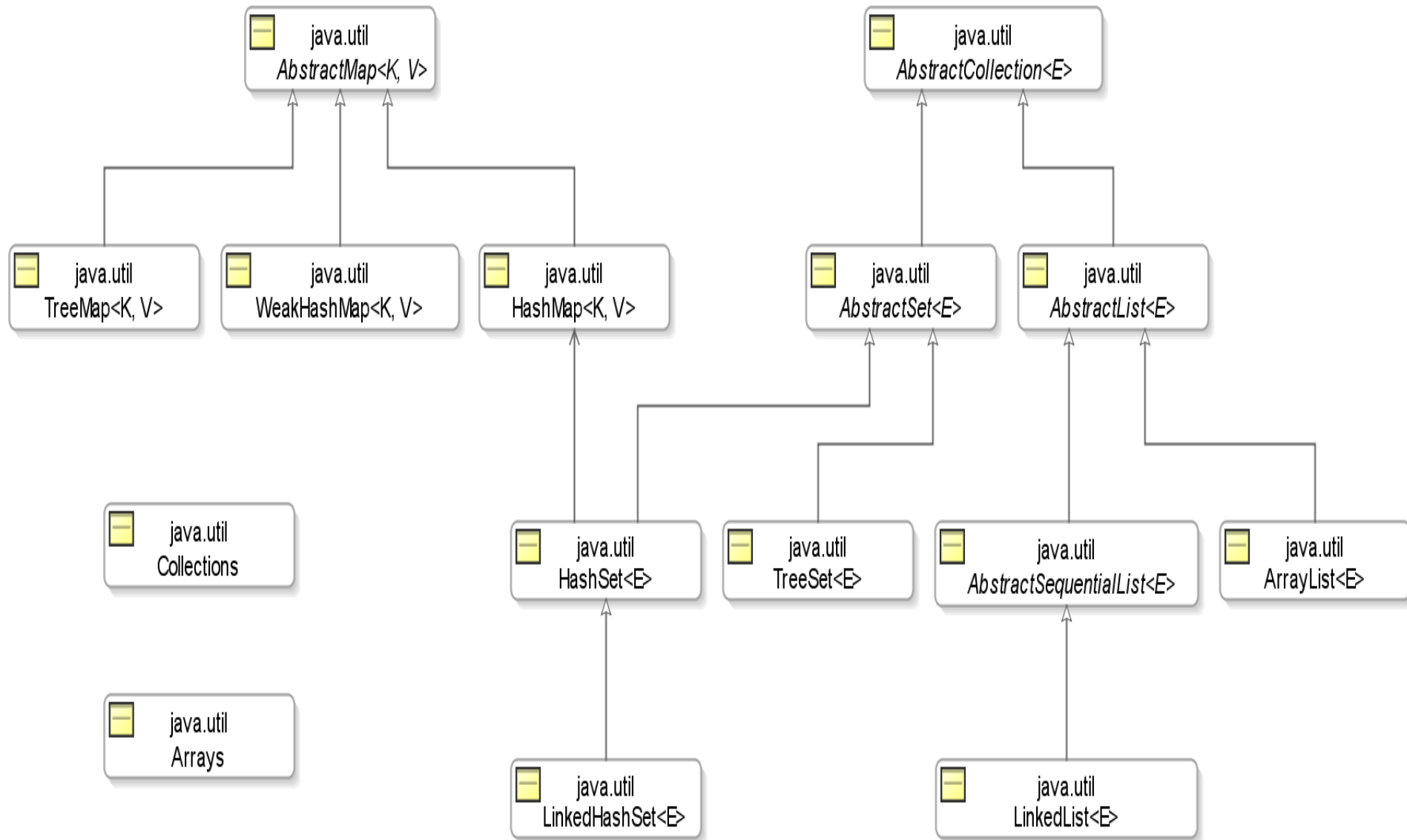
❖ Interfaces de Soporte

INTERFAZ	DESCRIPCIÓN
Iterator	Sustituye a la interfaz Enumeration . Dispone de métodos para recorrer una colección y para borrar elementos.
ListIterator	Deriva de Iterator y permite recorrer lists en ambos sentidos.
Comparable	Declara el método compareTo() que permite ordenar las distintas colecciones según un orden natural (String, Date, Integer, Double, ...).
Comparator	Declara el método compare() y se utiliza en lugar de Comparable cuando se desea ordenar objetos no estándar o sustituir a dicha interfaz.

Implementación del Core



Implementación del Core



Implementación del Core

❖ Clases de Propósito General

CLASE	DESCRIPCIÓN
HashSet	Implementación de la interfaz Set mediante una hash table.
TreeSet	Implementación de la interfaz SortedSet mediante un árbol binario ordenado.
ArrayList	Implementación de la interfaz List mediante un array.
LinkedList	Implementación de la interfaz List mediante una lista vinculada.
HashMap	Implementa la interfaz Map mediante una hash table.
WeakHashMap	Implementa la interfaz Map de modo que la memoria de los pares clave/valor pueda ser liberada cuando las claves no tengan referencia desde el exterior de la WeakHashMap .
TreeMap	Implementa la interfaz SortedMap mediante un árbol binario.

Manejo de Listas

❖ Creación de una lista

```
List<Producto> lista = new ArrayList<Producto>();
```

❖ Agregar Elementos

```
lista . add ( new Producto( "Televisor", 890.0, 256 ) );  
lista . add ( new Producto( "Lavadora", 1500.0, 435) );
```

❖ Recorrer los elementos de una lista

```
for ( Producto prod : Lista ) {  
    . . .  
}
```

Manejo de Datos tipo Clave / Valor

❖ Creación de una lista

```
Map<String,Producto> lista = new HashMap<String, Producto>();
```

❖ Agregar Elementos

```
lista . put("Televisor", new Producto("Televidor", 879.99, 450));  
lista . put("Lavadora", new Producto("Lavadora", 1435.67, 236));
```

❖ Recorrer los elementos de una lista

```
for( String clave: lista.keySet() ){  
    Producto prod = lista.get(clave);  
    . . .  
}
```

Bibliografía

- ❖ Lenguaje de Programación Java 2 Versión 5
Eric Gustavo Coronel Castillo
- ❖ Desarrollando Soluciones con Java y MySQL Server
Eric Gustavo Coronel Castillo
- ❖ Programación Java 2
Joel Carrasco Moñoz
- ❖ Piensa en Java
Bruce Eckel
- ❖ Java Programming Language - Student Guide
Sun Microsystems
- ❖ Como Programar en Java
Deitel y Deitel
- ❖ Java 2
Steven Holzner
- ❖ Aprende Java como si estuviera en primero
Javier García de Jalón et al