ESTRUCTURAS DE DATOS

NOTAS SOBRE JAVA

Contenedores lineales

Manuel Montenegro Montes
Departamento de Sistemas Informáticos y Computación
Facultad de Informática – Universidad Complutense de Madrid

Librería estándar de Java

- Proporciona un gran número de clases con implementaciones de los TADs más comunes, y algoritmos para su manipulación.
- Estas clases suelen estar en el paquete `java.util`.

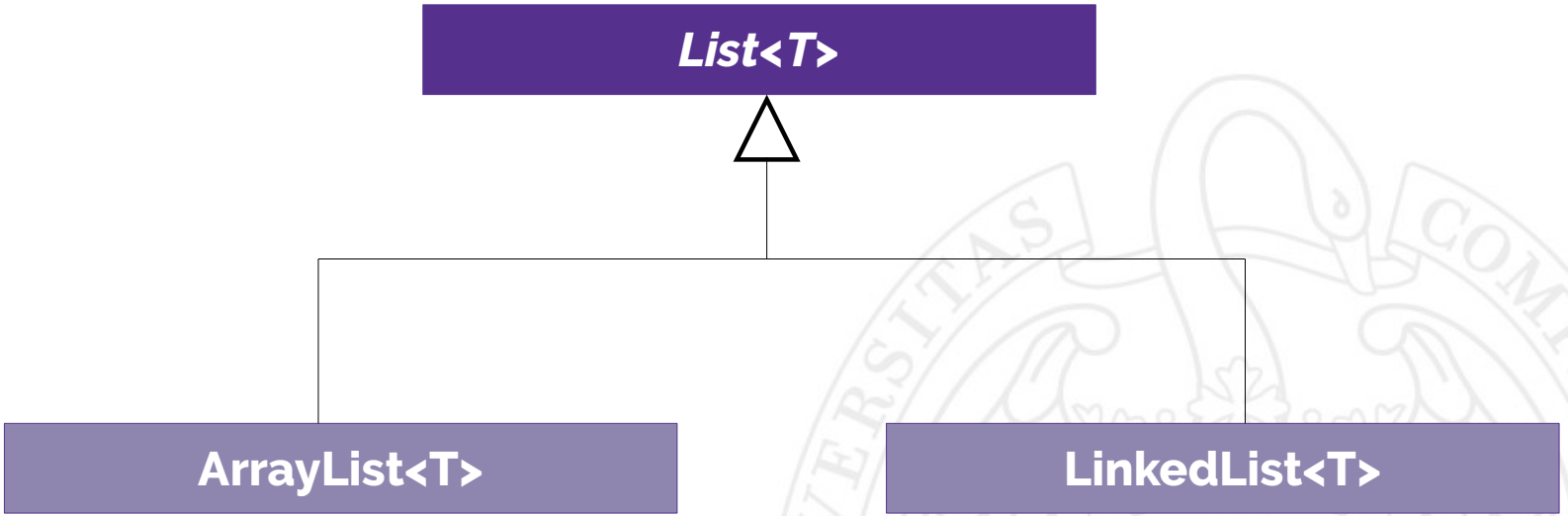
Clases de TAD lineales

La interfaz *List*

- Define las operaciones del TAD Lista.

List<T>
+ add(T) + add(pos, T) + remove(pos) + contains(T): bool + get(pos): T + set(pos, T) + size(): int + subList(pos1, pos2): List<T> + toArray(T[]): T[] ...

Implementaciones de `List<T>`

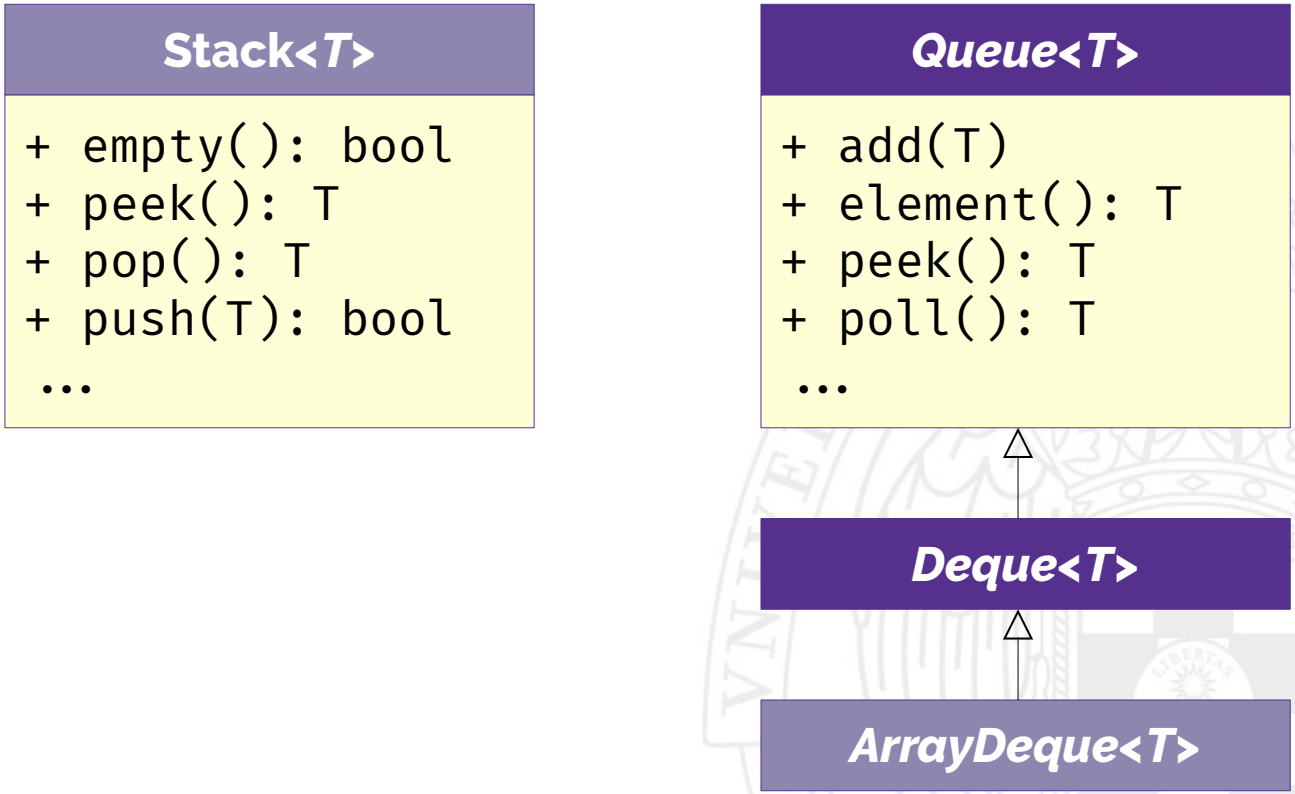


Ejemplo

```
List<Integer> l = new ArrayList<>();
for (int i = 0; i < 10; i++) {
  l.add(i * 3);
}
System.out.println(l);
```

```
[0, 3, 6, 9, 12, 15, 18, 21, 24, 27]
```

Otros TADs lineales

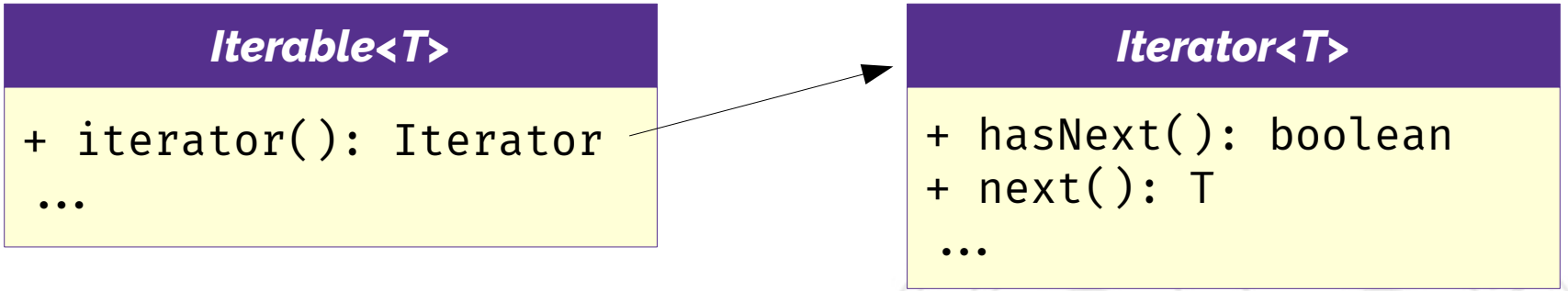


Iteradores

Iteradores

- Es posible obtener un iterador a partir de cualquier clase que implemente la interfaz `Iterable`.
  - `List`
  - `Stack`
  - `Queue`
  - etc.

Iteradores

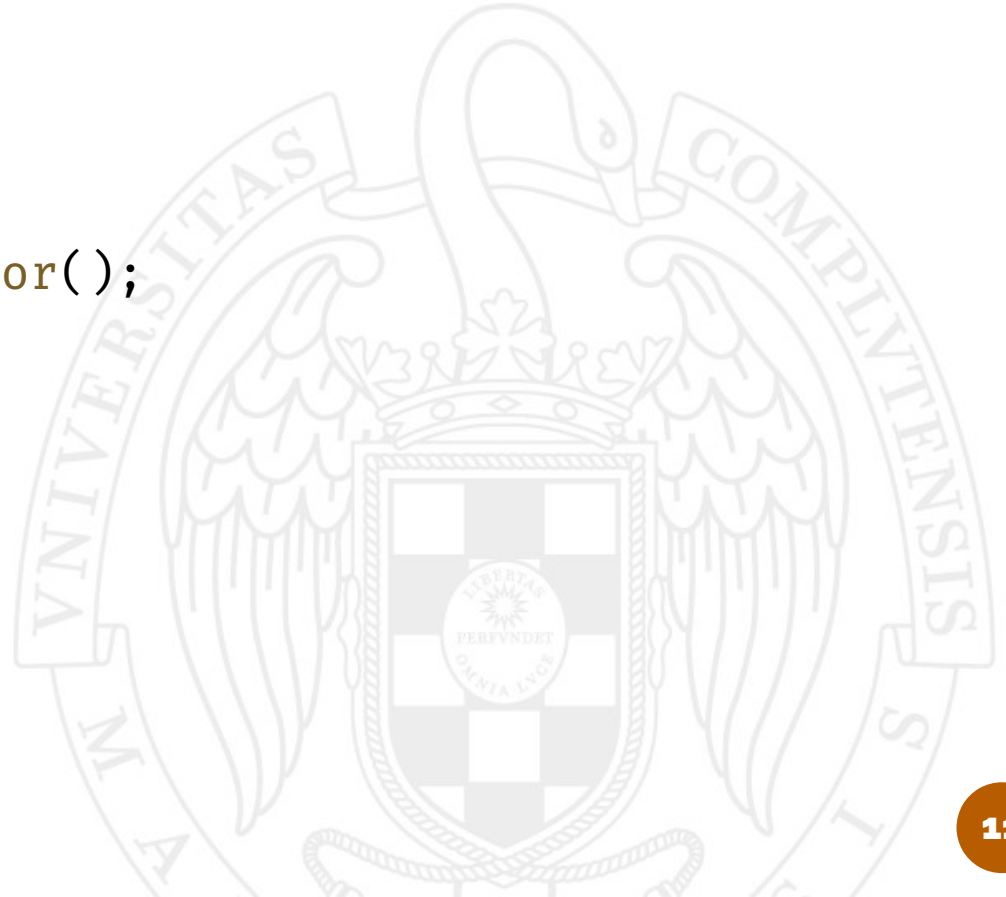

Iterable<T>
+ iterator(): Iterator
...
Iterator<T>
+ hasNext(): boolean
+ next(): T
...

Ejemplo

```
List<Integer> l =  ... ;

...  // Insertar elementos en l

int suma = 0;
Iterator<Integer> it = l.iterator();
while (it.hasNext()) {
  suma += it.next();
}
System.out.println(suma);
```



Sintaxis alternativa

```
List<Integer> l =  ... ;

...  // Insertar elementos en l

int suma = 0;
Iterator<Integer> it = l.iterator();
while (it.hasNext()) {
  suma += it.next();
}
System.out.println(suma);
```

=

```
for (Integer x: l) {
  suma += x;
}
```

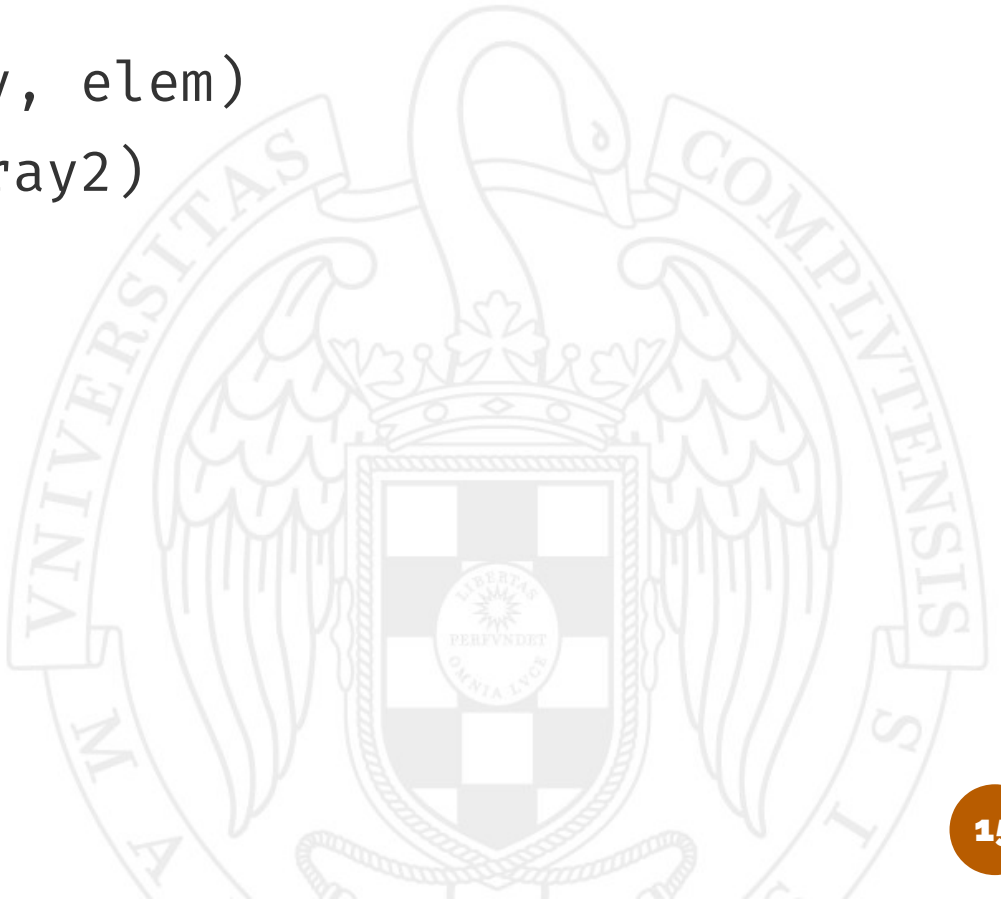
Funciones de utilidad

La clase `Collections`

- Contiene varios métodos estáticos que trabajan con listas.
  - `Collections.copy(list_dest, list_orig)`
  - `Collections.fill(list, elem)`
  - `Collections.max(list)`
  - `Collections.binarySearch(list, elem)`
  - `Collections.sort(list)`

La clase Arrays

- Utilidades similares, pero para arrays en lugar de listas.
  - `Arrays.asList(elems)`
  - `Arrays.binarySearch(array, elem)`
  - `Arrays.equals(array1, array2)`
  - `Arrays.sort(array)`
  - `Arrays.toString(array)`



Ejemplo

```
List<String> l = Arrays.asList("Ricardo", "Adrián", "Lucía", "Clara");
Collections.sort(l);
System.out.println(l);
```

```
[Adrián, Clara, Lucía, Ricardo]
```

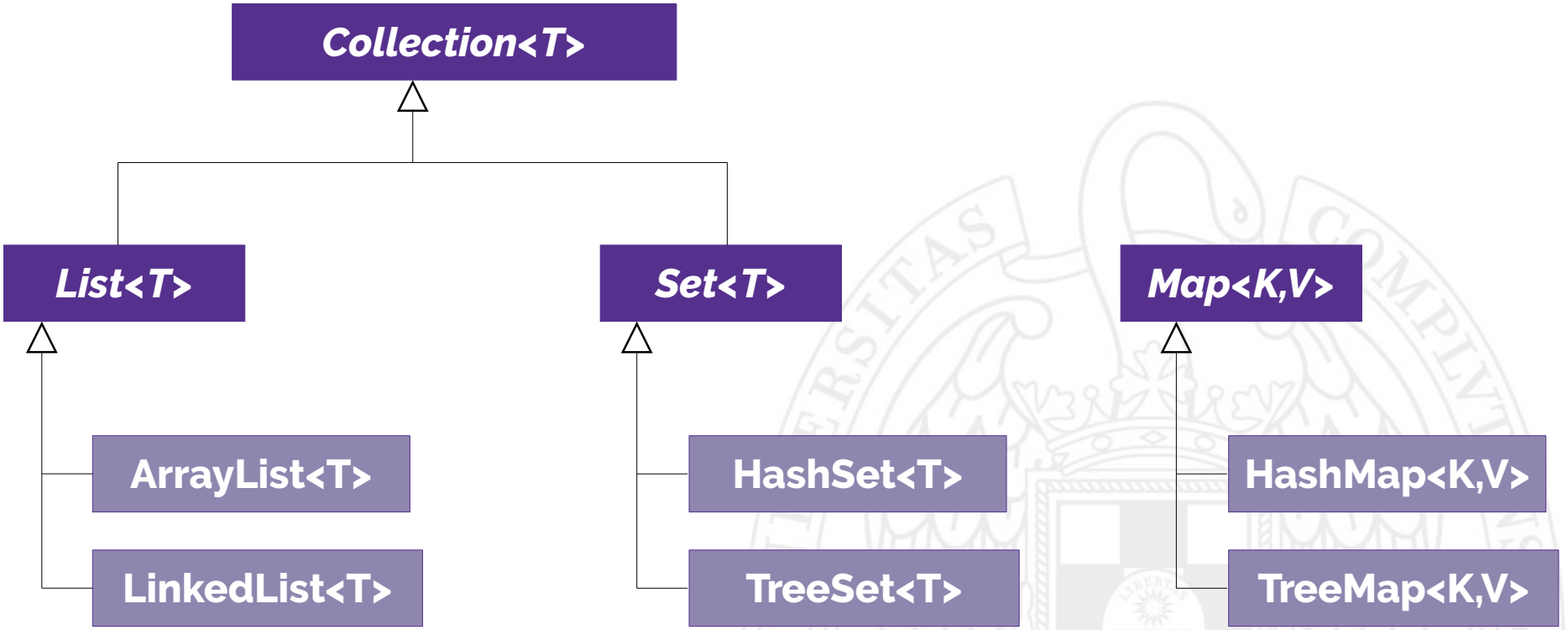
ESTRUCTURAS DE DATOS

NOTAS SOBRE JAVA

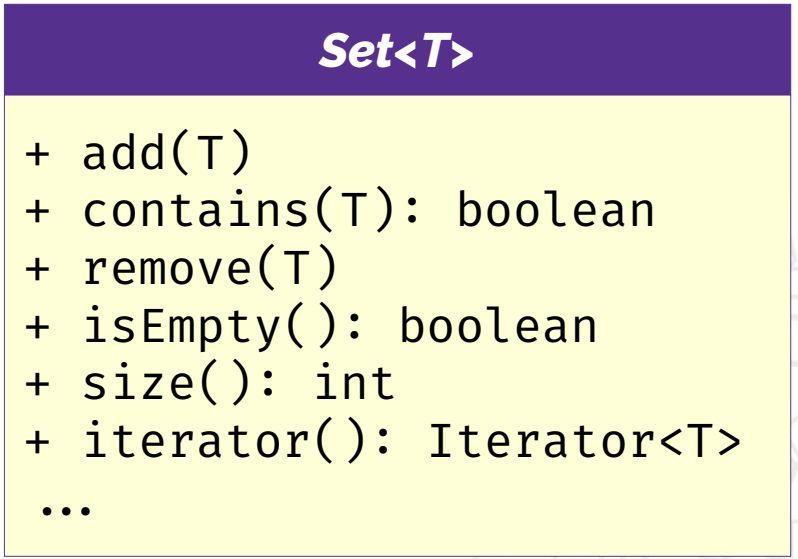
Contenedores asociativos

Manuel Montenegro Montes
Departamento de Sistemas Informáticos y Computación
Facultad de Informática – Universidad Complutense de Madrid

Jerarquía de contenedores



Interfaz Set



Interfaz Map

Map<K, V>

```
+ put(K, V)
+ containsKey(K): boolean
+ get(K): V
+ remove(K)
+ isEmpty(): boolean
+ size(): int
+ entrySet(): Set<Map.Entry<K,V>>
...
```

Comparación entre elementos

Comparable<T>
`+ compareTo(T): int`

`x.compareTo(y)` devuelve:

- < 0 si x < y
- = 0 si x == y
- > 0 si x > y

Object
`+ equals(Object): boolean` `...`

Ejemplo

```
public class Fecha implements Comparable<Fecha> {
  private int dia, mes, anyo;
  ...
  @Override
  public int compareTo(Fecha f) {
    if (anyo == f.getAnyo()) {
      if (mes == f.getMes()) {
        return dia - f.getDia();
      } else {
        return mes - f.getMes();
      }
    } else {
      return anyo - f.getAnyo();
    }
  }

  @Override
  public boolean equals(Object o) {
    if (o instanceof Fecha) {
      Fecha f = (Fecha)o;
      return f.getDia() == dia && f.getMes() == mes && f.getAnyo() == anyo;
    } else return false;
  }
}
```

Ejemplo

```
Set<Fecha> sf = new TreeSet<>();

sf.add(new Fecha(10, 4, 2010));
sf.add(new Fecha(3, 10, 2011));
sf.add(new Fecha(1, 10, 2010));

for (Fecha f: sf) {
  System.out.println(f);
}
```

```
10/04/2010
01/10/2010
03/10/2011
```

Funciones *hash*

Object

`+ equals(Object): boolean`
`+ hashCode(): int`
`...`

Si `x.equals(y)` devuelve `true`, entonces debe cumplirse `x.hashCode() == y.hashCode()`

Ejemplo

```
public class Fecha implements Comparable<Fecha> {
  private int dia, mes, anyo;
  ...
  @Override
  public int hashCode() {
    int result = anyo;
    result *= 100;
    result += mes;
    result *= 100;
    result += dia;
    return result;
  }
}
```

Ejemplo

```
Set<Fecha> sf = new HashSet<>();
sf.add(new Fecha(10, 4, 2010));
sf.add(new Fecha(10, 4, 2010));

for (Fecha f: sf) {
  System.out.println(f);
}
```

```
10/04/2010
```