

UD12- Interfaces

1. Qué es una interface.
2. Interface vs clase abstracta.
3. Definir una interface.
4. Herencia e implementación de interfaces.
5. Ejemplo.
6. Interfaces de Java: Comparable, serializable....

1. Qué es una Interface

Podemos considerar una interface como una **clase abstracta pura**.

Todos los métodos de una interface son públicos y abstractos aunque no se indique explícitamente. Por lo tanto, no es necesario escribir en cada método *public abstract*.

Una interface puede contener atributos constantes. **Las constantes declaradas son públicas y estáticas** aunque no se indique explícitamente. Por lo tanto, se pueden omitir los modificadores *public static final* cuando se declara el atributo. Las constantes se deben inicializar en la misma instrucción de declaración.

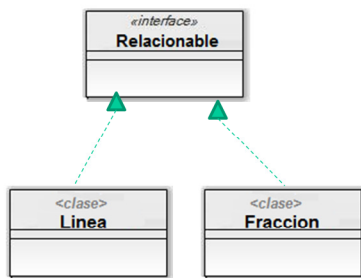
2. Interfaz vs. clase abstracta

- En la interface todo método es abstracto y público sin necesidad de declararlo. Una clase abstracta puede tener métodos abstractos y no abstractos.
- Los atributos declarados en una interface son public static final. Una clase abstracta puede contener también atributos de otro tipo (de instancia, no constantes, private, etc).

Las interfaces juegan un papel fundamental en la creación de aplicaciones Java ya que permiten interactuar a objetos no relacionados entre sí. Utilizando interfaces es posible que clases no relacionadas, situadas en distintas jerarquías de clases sin relaciones de herencia, tengan comportamientos comunes.

3. Definición de una interface

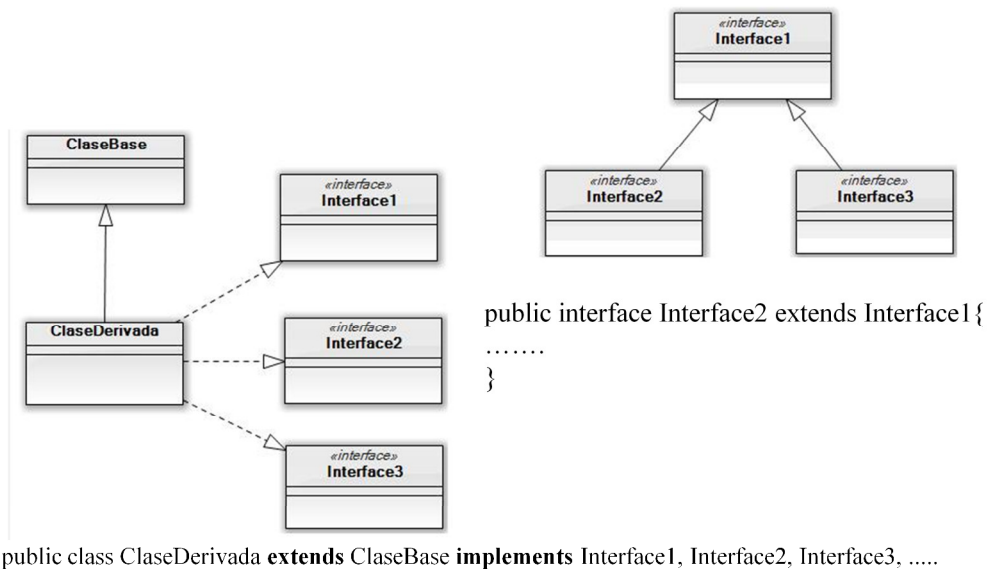
```
[public] interface nombreInterfaz [extends superInterface ]{  
    [atributos constantes]  
    [declaración de métodos abstractos]  
}
```



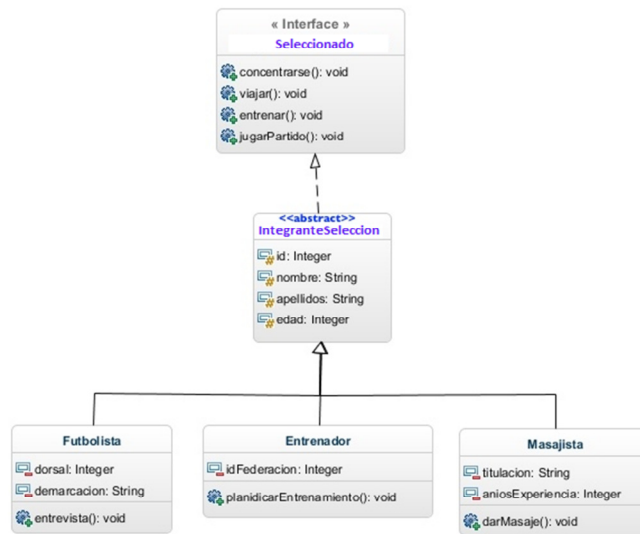
```
public interface Relacionable {  
    boolean esMayorQue(Object a);  
    boolean esMenorQue(Object a);  
    boolean esIgualQue(Object a);  
}
```

A diagram showing a line segment between two points labeled (x_1, y_1) and (x_2, y_2) . The formula for the length of the segment is given as $Longitud = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

4. Herencia/Implementación Interface



5. Ejemplo



6. Comparable, serializable....

java.lang

Interface Comparable<T>

Type Parameters:

T - the type of objects that this object may be compared to

Existen una serie de **convenciones para nombrar a los genéricos:**

E – Element (usado bastante por Java Collections Framework)

K – Key (Clave, usado en mapas)

N – Number (para números)

T – Type (Representa un tipo, es decir, una clase)

V – Value (representa el valor, también se usa en mapas)

S,U,V etc. – usado para representar otros tipos.

6. Comparable, serializable....

java.lang

Interface Comparable<T>

Type Parameters:

T - the type of objects that this object may be compared to

Method Summary

Methods

Modifier and Type	Method and Description
int	<code>compareTo(T o)</code> Compares this object with the specified object for order.

```
public class NClase implements Comparable<T>{  
}
```


6. Comparable, serializable....

Interface Comparable en Java 7

AEF Registros
Ud10Act1

Empleado
-nombre: String -apellidos: String -departamento: short -seccion: byte -sueldo: float
+Empleado(nombre:String, apellidos:String, departamento:short, seccion:byte, sueldo:float) +getters()

Empleado
-nombre: String -apellidos: String -departamento: short -seccion: byte -sueldo: float
+Empleado(nombre:String, departamento:short, seccion:byte, sueldo:float) +equals(obj:Object): boolean +toString(): String



<<Interface>> Comparable

sueldo

6. Comparable, serializable....

Clase Arrays en Java 7

Es una librería para trabajar con arrays ([])

```
java.util
Class Arrays
java.lang.Object
java.util.Arrays
```



Comparable

```
static void      sort(int[] a)  
Sorts the specified array into ascending numerical order.
```

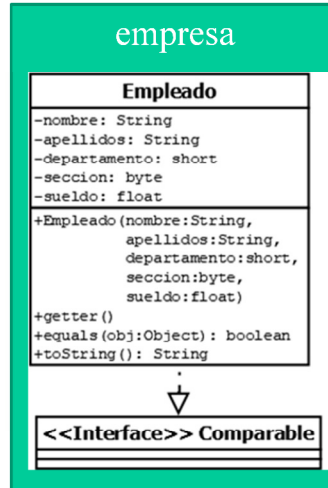
```
static void      sort(Object[] a)  
Sorts the specified array of objects into ascending order, according to the natural ordering  
of its elements.
```

```
static <T> void   sort(T[] a, Comparator<? super T> c)  
Sorts the specified array of objects according to the order induced by the specified  
comparator.
```

6. Comparable, serializable....

Hacer un programa que cree una tabla de 4 empleados y mostrar la tabla en **orden de introducción** y ordenada **por sueldo** (orden natural)

```
empresa=new Empleado[4];  
//cargamos  
Arrays.sort(empresa);  
//se ha reordenado con orden natural
```



6. Comparable, serializable....

Interface Comparator en Java 7

java.util

Interface Comparator<T>

Type Parameters:

T - the type of objects that may be compared by this comparator

Method Summary

Methods

Modifier and Type	Method and Description
int	compare (T o1, T o2) Compares its two arguments for order.
boolean	equals (Object obj) Indicates whether some other object is "equal to" this comparator.

6. Comparable, serializable....

Hacer un programa que cree una tabla de 4 empleados y mostrar la tabla en **orden de introducción**, ordenada en **orden natural**, ordenada por **departamento y nombre ascendente** y ordenada por **departamento ascendente y sección descendente**.

NOMBRE: Gutiérrez Gómez Juan DEPT: 20 SECC: 2 SUELDO: 1700.0
NOMBRE: Zoido Salma Antonio DEPT: 10 SECC: 1 SUELDO: 1500.0
NOMBRE: Abad Pedrosa Ana DEPT: 20 SECC: 1 SUELDO: 1300.0
NOMBRE: García Pérez Luisa DEPT: 10 SECC: 2 SUELDO: 1800.0

ORDENADO SUELDO

NOMBRE: Abad Pedrosa Ana DEPT: 20 SECC: 1 SUELDO: 1300.0
NOMBRE: Zoido Salma Antonio DEPT: 10 SECC: 1 SUELDO: 1500.0
NOMBRE: Gutiérrez Gómez Juan DEPT: 20 SECC: 2 SUELDO: 1700.0
NOMBRE: García Pérez Luisa DEPT: 10 SECC: 2 SUELDO: 1800.0

ORDENADO DEPARTAMENTO Y NOMBRE

NOMBRE: García Pérez Luisa DEPT: 10 SECC: 2 SUELDO: 1800.0
NOMBRE: Zoido Salma Antonio DEPT: 10 SECC: 1 SUELDO: 1500.0
NOMBRE: Abad Pedrosa Ana DEPT: 20 SECC: 1 SUELDO: 1300.0
NOMBRE: Gutiérrez Gómez Juan DEPT: 20 SECC: 2 SUELDO: 1700.0

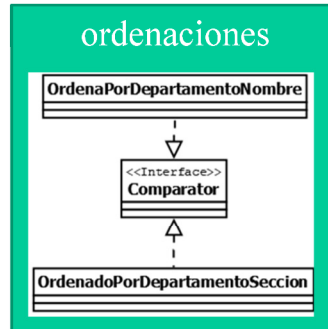
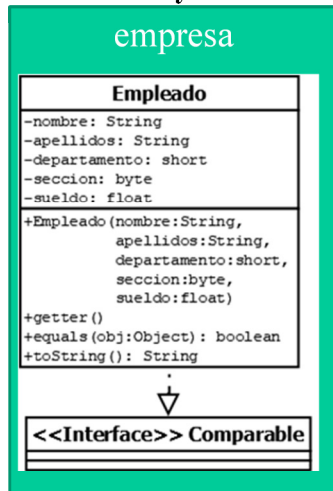
ORDENADO DEPARTAMENTO Y SECCIÓN DESCENDENTE

NOMBRE: García Pérez Luisa DEPT: 10 SECC: 2 SUELDO: 1800.0
NOMBRE: Zoido Salma Antonio DEPT: 10 SECC: 1 SUELDO: 1500.0
NOMBRE: Gutiérrez Gómez Juan DEPT: 20 SECC: 2 SUELDO: 1700.0
NOMBRE: Abad Pedrosa Ana DEPT: 20 SECC: 1 SUELDO: 1300.0

Programación Nuria Fuentes Pérez

6. Comparable, serializable....

Hacer un programa que cree una tabla de 4 empleados y mostrar la tabla en **orden de introducción**, ordenada en **orden natural**, ordenada por **departamento y nombre ascendente** y ordenada por **departamento ascendente y sección descendente**.



```
empresa=new Empleado[4];
//cargamos
Arrays.sort(empresa,new OrdenaPorDepartamentoNombre);
//se ha reordenado con orden marcado por Comparator
```