

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS



# Introducción a Ciencias de la Computación

## *Practica 20*

Profesora: Amparo López Gaona  
Ayudante: Diana Isabel Ramírez García  
Ayudante: Francisco Manuel Monreal Gamboa

## 1. Objetivo

El objetivo de esta práctica es que el alumno ejercite sus conocimientos acerca de la utilidad y programación de interfaces durante el desarrollo de sus programas. Las prácticas incluyen la implementación de interfaces proporcionadas por Java.

## 2. Introducción

Una *interfaz* puede verse como el caso extremo de una clase abstracta, pues en la interfaz todos los métodos son abstractos, es decir, sólo se define el comportamiento de las clases que las implementan, mediante los métodos sin especificar la implementación de ninguno.

La definición sintáctica de una interfaz se muestra a continuación:

```
visibilidad interface nombre_de_la_interfaz {  
    declaración de constantes y métodos  
}
```

La interfaz puede definirse pública o sin modificador de acceso, y tiene el mismo significado que para las clases. En el cuerpo de la interfaz aparecen constantes públicas y firmas de métodos públicos. Debido a que no tiene sentido utilizar cualquier otro calificador de visibilidad para los atributos y métodos de una interfaz, pueden omitirse y se asume que todo es público. Lo común es que las interfaces tengan un nombre con el sufijo **able**.

Para usar una interfaz se debe especificar en una clase que ésta la implementa mediante la palabra reservada **implements**. La clase que implementa la interfaz debe implementar al menos todos los métodos de ella, es decir, puede contener otros métodos. Resulta indispensable que la firma de cada método definido en la interfaz coincida con la firma del método en la clase que los implementa. De otra forma se tiene sobrecarga de métodos y el compilador envía un mensaje de error indicando que falta de implementarse un método.

### 3. Desarrollo

Durante esta práctica se desarrollará una clase para ordenar objetos de clases que implementen la interfaz `Comparator` del paquete `java.util` y se programarán varias implementaciones de tal interfaz.

1. Descargar los archivos `Punto.class`, `Punto.html`, `Persona.class`, `Persona.html` y `PruebaOrden.java` en un mismo directorio.
2. Escribir una clase `PuntoComparable` que implemente la interfaz `Comparator` y permita determinar la relación de orden entre dos puntos. Para comparar dos puntos calcular la distancia al origen de cada uno y el punto que esté más cerca del origen es menor que el otro.
3. Escribir una clase `PersonaComparable` que implemente la interfaz `Comparator` que permita comparar dos personas. Una persona es mayor que otra si su edad es mayor y en caso de empate se considera la estatura.
4. Probar la implementación de las clases ejecutando el programa `PruebaOrden`.

### Salida:

Los puntos que se van a ordenar son:

(19,4) (-3,2) (10,10) (89,8) (-2,-10) (34,56)

Los puntos ordenados son

(-3,2) (-2,-10) (10,10) (19,4) (34,56) (89,8)

Las personas que se van a ordenar son:

Maria 15 1.5

Daniela 15 1.51

Alejandra 15 1.48

Andrea 19 1.57

Blanca 23 1.56

Jorge 23 1.7

Paco 26 1.8

Mariana 8 1.2

Diego 8 1.22

Mario 3 0.9

Las personas ordenadas por edad y estatura son:

Mario 3 0.9

Mariana 8 1.2

Diego 8 1.22

Alejandra 15 1.48

Maria 15 1.5

Daniela 15 1.51

Andrea 19 1.57

Blanca 23 1.56

Jorge 23 1.7

Paco 26 1.8

## 4. Forma de entrega

1. Las prácticas serán entregadas de forma individual.
2. Cada práctica (sus archivos y directorios) deberá estar contenida en un directorio llamado `<apellidoPaterno_nombre_pX>`, donde X es el número de la práctica.  
Por ejemplo: `monreal_francisco_p1`
3. **NO** incluir los archivos `.class` dentro de la carpeta
4. Comprimir el directorio `<apellidoPaterno_nombre_pX>`, en un archivo `<apellidoPaterno_nombre_pX>.zip`  
Por ejemplo: `monreal_francisco_p1.zip`
5. Las prácticas deberán ser entregadas durante la clase de laboratorio, la hora límite será a las 12:00 del día en que se publiquen.
6. Los archivos de código fuente deben estar documentados.
7. Se pueden discutir y resolver dudas entre los integrantes del grupo. Pero cualquier práctica plagiada total o parcialmente será penalizada con cero para los involucrados.
8. La práctica sera enviada al classroom de la clase en la sección que le corresponda.