

## Tarea 5 – Abstracción e interfaces

### Enunciado

1. Diseña una clase abstracta con un constructor que imprime "Este es un constructor de clase abstracta", un método abstracto llamado `abstract_method` y un método no abstracto que imprime "Este es un método normal de clase abstracta". Una clase `'SubClase'` hereda la clase abstracta y tiene el método llamado `abstract_method` que imprime "Este es un método abstracto implementado en una clase no abstracta". Ahora crea un objeto de `SubClase` y llame al método abstracto y al método no abstracto. (Analizar el resultado).
2. Diseña una jerarquía de clases que empiece en la clase `Employee` que incluya subclases para `HourlyEmployee` y `SalaryEmployee`. Los atributos compartidos por estas clases incluyen el `nombre` y el `cargo` del empleado, además de los métodos getter y setter necesarios para esos atributos. Los empleados asalariados necesitan un atributo para el salario semanal y los métodos correspondientes para acceder y cambiar esta variable. Los empleados por hora deben tener una tasa de pago y una variable de horas trabajadas. Debería haber un método abstracto llamado `"calculateWeeklyPay()"`, definido de forma abstracta en la superclase e implementado en las subclases. La paga del trabajador asalariado es solo el salario semanal. El pago de un empleado por hora es simplemente la tarifa de pago por horas trabajadas.
3. Responde a las siguientes preguntas:
  - a. ¿Una clase abstracta posee constructor? Si es así, ¿se podría crear un objeto de una clase abstracta?
  - b. ¿Cuáles son las diferencias entre una clase abstracta y una interfaz?
  - c. ¿Puede una clase abstracta implementar una interfaz? Razona tu respuesta.
4. Crea una clase abstracta `Forma`:
  - a. – Atributos: `numLados`
  - b. – Constructor: inicializar `numLados`
  - c. – Método: getter para `numLados`
  - d. – Métodos abstractos: `getArea()`, `getPerimeter()`Escribir una subclase `Rectángulo`
  - e. – Atributos: `ancho`, `alto`Escribe una subclase `Triángulo`
  - f. – Atributos: `ancho`, `alto`En otra clase, escriba un método `main` para definir un `Rectángulo` y un `Triángulo`.
5. Crea una interfaz `Redimensionable`
  - a. – Tiene un método `redimensionar(int x)` que cambia el tamaño de las dimensiones de una `Forma` por el factor `x`Hacer que `Rectángulo` implemente `Redimensionable`  
Escriba un método `main` para:
  - b. - Definir un `Rectángulo` (`ancho = 2`, `altura = 3`)
  - c. - Imprime el área y el perímetro del `Rectángulo`
  - d. - Cambiar el tamaño del `Rectángulo` por factor de 2
  - e. - Vuelva a imprimir el área y el perímetro del `rectángulo`

## UD5. Herencia y polimorfismo

6. La aplicación dada debería ordenar 10 objetos que representan 10 Rectángulos **según su área**. Para hacer que esta aplicación se ejecute, la clase Rectángulo debe implementar la interfaz `java.lang.Comparable` (es una interfaz que se puede usar para ordenar los elementos de la colección.). Modifica la clase Rectángulo creada en el ejercicio anterior, sobrescribe el método `toString` y haz que implemente la interfaz `Comparable<Rectángulo>` y añade el código necesario para comparar dos objetos rectángulos y que funcione el siguiente código.  
Nota: leer para entender cómo funciona la interfaz `Comparable`.  
[https://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=587:ejercicio-ejemplo-resuelto-interface-comparable-y-metodo-compareto-api-java-comparar-objetos-cu00913c&catid=58&Itemid=180](https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=587:ejercicio-ejemplo-resuelto-interface-comparable-y-metodo-compareto-api-java-comparar-objetos-cu00913c&catid=58&Itemid=180)
7. Crea una clase Estudiante con atributos nombre, edad y altura. Queremos ordenar por edad y por altura a los estudiantes. Utilizando `Comparable` y el método `compareTo` debemos comparar los estudiantes y ordenarlos por altura primero (de mayor a menor) y si la altura coincide por edad. Por pantalla debe mostrarse la lista de estudiantes sin ordenar, y ordenado. Ejemplo:

Estudiantes sin ordenar:

1. Patri - Altura: 170 - Edad: 12
2. Manuel - Altura: 173 - Edad: 43
3. Javier - Altura: 189 - Edad: 72
4. Alicia - Altura: 168 - Edad: 52
5. Alberto - Altura: 189 - Edad: 35

Estudiantes ordenados:

1. Javier - Altura: 189 - Edad: 72
2. Alberto - Altura: 189 - Edad: 35
3. Manuel - Altura: 173 - Edad: 43
4. Patri - Altura: 170 - Edad: 12
5. Alicia - Altura: 168 - Edad: 52

## Entrega

- Sube un PDF con las respuestas y el código copiado.