

Programación

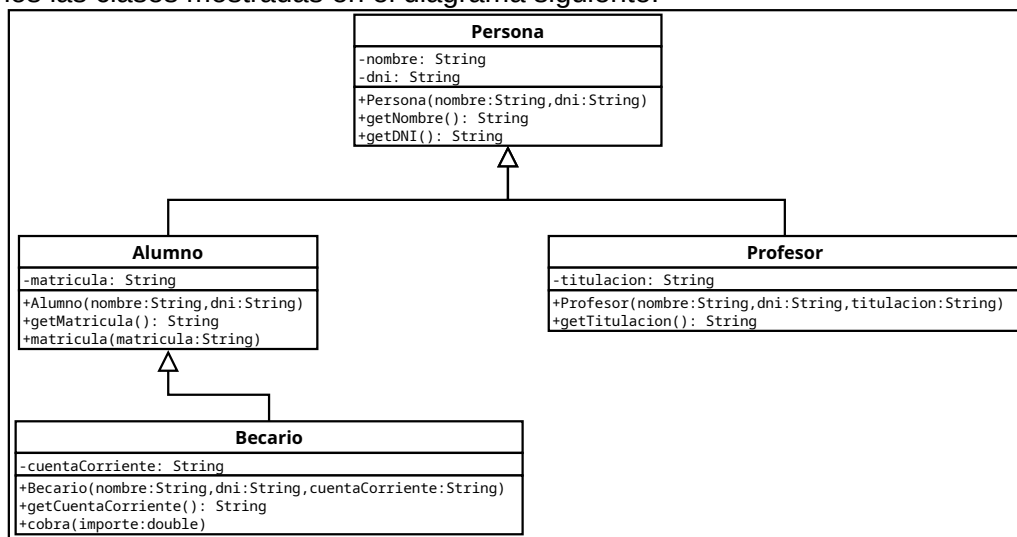
Bloque 04 - Desarrollo de Clases

Herencia e Interfaces

Todas las clases correspondientes a los ejercicios de esta relación se deben crear dentro del paquete `prog.unidad04.actividad405`, creando sub-paquetes de nombre `ejercicio01`, `ejercicio02`, etc., uno por ejercicio de la relación.

Asimismo hay que documentar todas las clases usando el formato Javadoc

1. Tenemos las clases mostradas en el diagrama siguiente:



Las clases se proporcionan en el archivo `Actividad_4.05a.jar` y la documentación de las mismas se proporciona en el archivo `Actividad_4.05a_doc.zip`

Crea una clase, llamada `prog.unidad04.actividad405.ejercicio01.Universidad`, que ofrezca e implemente los siguientes **métodos estáticos**.

- `void imprimir(Persona persona)`. Debe imprimir **toda la información de que se disponga de la persona** que se proporciona por parámetros. Esto significa que si la persona es un Profesor, se debe imprimir su titulación, por ejemplo.
- `void matricular(Object objeto, String matricula)` que debe matricular al objeto que se pase por primer parámetro con la matricula que se proporciona, si es posible. Tanto si lo es, como si no, debe imprimir por consola un mensaje indicando que se ha podido hacer, o no.

Utilizar el programa proporcionado en el archivo `Actividad_4.05a_pruebas.zip` para probar que tu clase funciona correctamente.

2. Crea el interfaz `prog.unidad04.actividad405.ejercicio02.Figura` con los siguientes métodos:
 - `double area()`. Devuelve el área de la figura
 - `double perimetro()`. Devuelve el perímetro de la figura.
 - `boolean esRegular()`. Devuelve si la figura es regular (todos los lados son iguales) o no.

Partiendo de este interfaz crea tres clases que lo implementen: `Circulo`, `Rectangulo` y `Triangulo`, en el mismo paquete que `Figura` sabiendo que un `Circulo` siempre es regular.

Emplea el programa en el archivo `Actividad_4.05b_pruebas.zip` para probar que tu ejercicio es correcto.

3. Vamos a realizar un sistema de cifrado.

Para ello vamos a definir el interfaz

`prog.unidad04.actividad405.ejercicio03.Cifra`, con los siguientes métodos:

- `String cifra(String mensaje)`. Cifra el mensaje obteniendo un mensaje cifrado.
- `String descifra(String mensajeCifrado)`. Descifra el mensaje cifrado, obteniendo el mensaje original.

Una vez creado este interfaz, crea la clase

`prog.unidad04.actividad405.ejercicio03.CifraCesar` que implemente el interfaz `Cifra`. Esta clase realiza el cifrado usando el método Cesar. El método de cifra Cesar usa como clave un número entero `n` (positivo o negativo) y para cifrar sustituye cada letra del mensaje original por la letra situada en el alfabeto `n` posiciones a la derecha (si es positivo) o `n` a la izquierda (si es negativo). Los caracteres que no sean letras minúsculas no se sustituyen por simplicidad y se mantendrán tal y como están. Para descifrar se usa el método inverso, sustituir cada carácter del mensaje cifrado por el situado en el alfabeto `n` posiciones a la izquierda (si es positivo) o `n` posiciones a la derecha (si es negativo). Debe tener también un constructor `CifraCesar(int clave)`. Asegurate que la clave está entre `-27` y `27`. Si no lo está se debe lanzar una excepción de tipo `IllegalArgumentException`.

Prueba tu programa con el programa contenido en el archivo

`Actividad_4.05c_pruebas.zip` y corrige lo que no funcione.

4. Crea la clase

`prog.unidad04.actividad405.ejercicio03.CifraPolialfabetica` que implemente la cifra polialfabética. En este caso la clave es una cadena con las letras del alfabeto, en minúsculas, pero en distinto orden al que aparecen en el mismo. Si la clave no contiene todas las letras minúsculas del alfabeto se deberá producir una excepción `IllegalArgumentException` al construir el objeto. Para cifrar se sustituye cada letra del mensaje por aquella que ocupa su posición en la clave. Para descifrar se sustituye cada letra por la que ocuparía la posición en la que se encuentre en la clave.

Prueba con el programa contenido en el archivo

`Actividad_4.05d_pruebas.zip` y corrige los problemas que aparezcan.