

Ejercicio Propuesto

Colecciones

Objetivo

- El objetivo de este ejercicio es afianzar los conocimientos sobre:
 - Manejo del API de Java
 - Conversión entre texto y fechas
 - Colecciones
 - Ordenación

PRIMERA PARTE:

Tratamiento de fechas

- Se debe crear una clase Persona, que tenga los siguientes atributos:
 - NIF (String)
 - fechaNacimiento (java.util.Date)

- Además tendrá los siguientes métodos:
 - Un constructor que tenga como argumentos el NIF y una cadena de texto (String) para la fecha de nacimiento en formato “31/12/2015” (día/mes/año). Dentro del constructor se deberá convertir la fecha de String a Date, usando la clase `java.text.SimpleDateFormat`, y asignar el resultado al atributo `fechaNacimiento`.
 - NOTA: el constructor deberá convertir de texto a fecha usando el método *parse* de la clase `SimpleDateFormat`. Este método lanza una excepción si el formato de fecha no es correcto. Puesto que aún no se ha visto el tema de excepciones, de momento lo único que el alumno debe hacer es declarar que este constructor puede lanzar esta excepción de la siguiente manera:

```
public Persona(...) throws ParseException {  
    ...  
}
```

- Otros métodos de la clase Persona:
 - `public String toString()`: devuelve una representación textual que nos permita ver los datos del objeto, incluyendo el NIF y la fecha de nacimiento en formato “31/12/2015” (día-mes-año). Para dar formato a la fecha nuevamente se debe utilizar `SimpleDateFormat` (método *format*).
 - Getters para los atributos:
 - `public String getNIF()`
 - `public Date getFechaNacimiento()`

Orientaciones

- El atributo fechaNacimiento es de tipo Date:
 - El constructor debe convertir de texto a fecha, ya que el argumento recibido es de tipo String
 - El método toString debe convertir de fecha a texto, ya que el valor está almacenado en el atributo que es de tipo Date
- Para convertir de texto a fecha o de fecha a texto utilizaremos SimpleDateFormat (métodos *parse* y *format*). Se debe consultar el API de java para saber cómo funciona esta clase:

<http://docs.oracle.com/javase/8/docs/api/java/text/SimpleDateFormat.html>

- No confundir minutos con meses

SEGUNDA PARTE: ORDENACIÓN NATURAL

- Definimos el orden natural como el orden alfabético por NIF.
- La clase Persona debe implementar el interfaz Comparable, y por tanto contendrá un método *compareTo* que determinará su orden natural, que en nuestro caso hemos definido como orden alfabético por NIF.
- Además se debe implementar el método *equals* de forma que sea coherente con esta definición de orden natural, es decir, dos objetos de la clase Persona serán iguales si y sólo si son iguales sus NIFs, sin tener en cuenta la fecha de nacimiento.
- Se deberá crear una lista de personas y ordenarla para probar que la operación funciona correctamente. Para ello se utilizará una clase aparte, OperaPersona, de la cual se facilita el esqueleto.

Orientaciones

```
public class Persona implements Comparable<Persona>{  
    ...  
    public int compareTo(Persona otraPersona) {  
        ...  
    }  
    public boolean equals(Object obj) {  
        ...  
    }  
}
```

- Para ordenar alfabéticamente cadenas de texto podemos apoyarnos en el método *compareTo* de la clase `String`.
- Para comparar cadenas de texto podemos apoyarnos en el método *equals* de la clase `String`.
- Consultar el API de la clase `String` para ver cómo funcionan estos métodos.

TERCERA PARTE:

ORDENACIÓN NO NATURAL

- Se creará otra lista de personas y, sin modificar los métodos equals ni compareTo de la clase Persona, se ordenará en base a la fecha de nacimiento, ascendentemente.
- Para ello se debe utilizar un Comparator, que se puede implementar en otra clase aparte o bien como clase anónima dentro de OperaPersona.
- Para ver si una fecha es anterior a otra podemos apoyarnos en el método *compareTo* de la clase `java.util.Date`. Consultar el API de esta clase para ver lo que hace este método.

CUARTA PARTE (OPCIONAL): CÁLCULO DE LA EDAD

- El alumno que lo desee podrá realizar esta funcionalidad. No puntuará para la nota, pero los tutores lo corregiremos para el que tenga interés.
- Se trata de crear un método en la clase Persona que calcule la edad a partir de la fecha de nacimiento y la fecha actual:

```
public int obtenerEdad(): obtiene la edad en años.
```
- La edad no debe ser un atributo del objeto Persona, sino que debe calcularse cada vez que se invoque este método (la edad va cambiando, no es un atributo fijo de la Persona; lo que no varía es su fecha de nacimiento que sí es un atributo)
- Podemos utilizar la clase `java.util.Calendar`. Se debe consultar el API de java para saber cómo funciona esta clase:
<http://docs.oracle.com/javase/8/docs/api/java/util/Calendar.html>