**Caso práctico de la unidad 3.**

**Hecho por Daniil Kobzar**

**15/12/2022**

Durante la creación de mi programa había creado 3 archivos .java

* El primero App.java con un método main donde vamos a ejecutar el programa y crear 2 objetos de la clase Persona
* El segundo Persona.java que sería una clase normal (no abstracta, ya que no vamos a tener jerarquía haciendo este ejercicio) con sus atributos con el acceso privado (utilizando correctamente el encapsulamiento) String nombre, String apellidos, String fechaCumple, int edad, String sexo. Ya que el usuario tiene que introducir los datos a mano he pensado en no crear el método constructor sino modificar un poco los setters a mi gusto.

Todos los getters cumplen su función con normalidad (devuelven el valor pedido por el usuario)

Por el otro lado en los setters había utilizado la clase Scanner. Para que el usuario pueda introducir los datos por el teclado. Sin embargo, hay 2 setters que destacan:

* + setFechaCumple – en el cual creo 2 arrays distintos. En el primero se almacena toda la información introducida por el usuario dividía por el símbolo “/”. En el segundo se almacena solo el mes y ahora explico por qué. Hay personas que introducen las fechas con un cero delante (05/05/2002). Para no cargar mucho el método que comprueba en que estación del año ha nacido la persona indicada he decidido hacer una comprobación en el mismo método setter. Por lo tanto, en el caso de que la persona había introducido el mes con un “0” delante se almacenara sin dicho “0”. Ejemplo: La persona introduce 04/04/2001 pues se almacenará como 04/4/2001.
  + setEdad – En este método en vez de utilizar scanner.nextInt() sigo utilizando scanner.nextline() y luego hago una comprobación utilizando el try catch de que la persona ha introducido los datos correctos. Lo hago de esta manera porque el método de scanner.nextint() después de leer una línea deja el cursor en la misma línea y pueden haber problemas como cuando el Scanner salta el input del usuario y no le deja introducir nada.
* Por último el interfaz MethodInterface.java el cual sería simplemente una interfaz con un método el cual posteriormente implementara la clase Persona utilizando la sobreescritura(@Override)
  + fechaDeNacimiento – es un método en el cual creo un array de la fecha de nacimiento dividido por “/”. Posteriormente hago una comprobación de la estación en la cual nació la persona creada. Lo envolví todo en un try catch por si la persona había introducido mal la fecha de nacimiento para que el programa no se cierre con un error de IndexOutOfBoundException. Aun así el apartado de “else” en el “try” comprueba por segunda vez si la fecha de nacimiento está mal escrita y no se puede decir la estación en la cual nació la persona