IES de Teis
Avda. de Galicia, 101
36216 – Vigo
Tfno: 886 12 04 64
e-mail: ies.teis@edu.xunta.es
http://www.iesteis.es

PRÁCTICA 1: COMPONENTES JAVA

Objetivos:

- Iniciar al alumnado en el uso de la interface serializable
- Iniciar al alumnado en el diseño de componentes con javaFX

Recursos:

Timer Class

Clase de temporizador proporciona la facilidad para que los subprocesos programen tareas para su ejecución futura en un subproceso en segundo plano. Las tareas se pueden programar para una ejecución única o para una ejecución repetida a intervalos regulares. los schedule() y scheduleAtFixedRate() Los métodos programan la tarea especificada para la ejecución repetida con demora fija y la ejecución con tasa fija, respectivamente. Estos métodos están sobrecargados.

El siguiente código utiliza el Timer clase para ejecutar el método ejecutar cada segundo (en Java 8 y superior):

```
import java.util.Timer;
import java.util.TimerTask;

class Main
{
   public static void main(String[] args) throws InterruptedException
   {
      Timer timer = new Timer();
      timer.schedule(new TimerTask() {
         @Override
         public void run() {
            System.out.println("Running: " + new java.util.Date());
         }
      }, 0, 1000);
   }
}
```

Secuencia y desarrollo:

1. **Ejercicio 1: Vamos a crear un reloj digital que poder insertar en cualquier interfaz, el reloj debe tener al menos las siguientes características:**
   - Una propiedad Timer
   - Una propiedad LocalTime
   - Una propiedad booleana para indicar si el formato es de 12 o 24 horas.
   - Método para dar formato de 12 o 24 horas
   - Método para iniciar el reloj
   - Método para parar el reloj
   - Debes crear dos constructores, uno sin parámetros y el otro con parámetros
   - Todos los atributos son privados

IES de Teis
Avda. de Galicia, 101
36216 – Vigo
Tfno: 886 12 04 64
e-mail: ies.teis@edu.xunta.es
http://www.iesteis.es

- Implementar todos los métodos setter y getter

2. **Ejercicio 2:Añadir al reloj digital las siguientes características y funcionalidad :**

   - Una propiedad booleana para indicar si queremos activar una alarma. El funcionamiento de la alarma consistirá en que se podrá configurar el componente para que a una determinada hora nos muestre un mensaje.
   - Dos propiedades para determinar la hora y minuto para el cual queremos programar la alarma. Ambas propiedades será de tipo entero.
   - Una propiedad para configurar el mensaje de texto, que queremos que se muestre, cuando se produzca el salto de la alarma. Esta propiedad será de tipo texto (String).
   - Función de alarma, si se programa a una hora, debe generar un evento cuando se llegue a esa hora.

3. **Ejercicio 3: Guardar el reloj digital en el disco duro para que pueda ser recuperado por otra interfaz. Para ello debes implementar la interface serializable e implementar el flujo de entrada y salida**

4. **Ejercicio 4: Crear una nueva interfaz en la que se muestre el reloj digital creado**