

Paradigmas de Programación

López Hernández Joshua Alejandro

Profesor: Cuauhtemoc Hidalgo Cortes

T.M ICO 3°

Clase Tiempo1:

```
J Tiempo1.java > Tiempo1 > toString()
1
2 //La clase Tiempo1 mantiene la hora en formato de 24 horas.
3 //Proporciona métodos para establecer y mostrar la hora en diferentes formatos.
4
5 public class Tiempo1 {
6     //Variables de instancia privadas para almacenar la hora
7     private int hora; //Almacena la Hora (0-23)
8     private int minuto; //Almacena los Minutos (0-59)
9     private int segundo; //Almacena los Segundos (0-59)
10
11
12     //Establece el tiempo verificando que los valores estén en rangos válidos
13     //Hora el valor de la hora (0-23)
14     //Minuto el valor de los minutos (0-59)
15     //Segundo el valor de los segundos (0-59)
16     //IllegalArgumentException si algún parámetro está fuera de rango
17
18     public void establecerTiempo(int hora, int minuto, int segundo) {
19         //Validación de rangos para hora, minuto y segundo
20         if (hora < 0 || hora >= 24 || minuto < 0 || minuto >= 60 || segundo < 0 || segundo >= 60) {
21             throw new IllegalArgumentException(
22                 s:"hora, minuto y/o segundo estaban fuera de rango");
23         }
24
25         //Asignación de valores validados a las variables de la instancia
26         this.hora = hora;
27         this.minuto = minuto;
28         this.segundo = segundo;
29     }
30
31     //Devuelve la Hora en formato universal de 24 Horas (HH:MM:SS) como String
32
33     public String aStringUniversal() {
34         return String.format(format:"%02d:%02d:%02d", hora, minuto, segundo);
35     }
36
37
38     //Devuelve la hora en formato estándar de 12 Horas (H:MM:SS) como String
39
40     public String toString() {
41         return String.format(format:"%d:%02d:%02d %s",
42             ((hora == 0 || hora == 12) ? 12 : hora % 12),
43             minuto, segundo, (hora < 12 ? "AM" : "PM"));
44     }
45 }
```

Clase PruebaTiempo1:

```
J PruebaTiempo1.java > PruebaTiempo1
1 //Case de prueba para demostrar el uso de la clase Tiempo1
2
3 public class PruebaTiempo1 {
4     Run | Debug
5     public static void main(String[] args) {
6         //Crear un objeto Tiempo1
7         Tiempo1 tiempo = new Tiempo1();
8         mostrarTiempo(encabezado:"Después de crear el objeto tiempo", tiempo);
9
10        //Establecer un tiempo válido
11        tiempo.establecerTiempo(hora:13, minuto:27, segundo:6);
12        mostrarTiempo(encabezado:"Después de llamar a establecerTiempo", tiempo);
13        System.out.println();
14
15        //Intentar establecer un tiempo inválido (A excepciones)
16        try {
17            tiempo.establecerTiempo(hora:99, minuto:99, segundo:99);
18        } catch (IllegalArgumentException e) {
19            System.out.printf(format:"Exception: %s%n%n", e.getMessage());
20        }
21
22        //Mostrar el tiempo después del intento fallido
23        mostrarTiempo(encabezado:"Después de llamar a establecerTiempo con valores invalidos", tiempo);
24    }
25
26
27    //Método auxiliar para mostrar el tiempo en ambos formatos
28    //Encabezado texto descriptivo para la salida
29    //t objeto Tiempo1 a mostrar
30
31    private static void mostrarTiempo(String encabezado, Tiempo1 t) {
32        System.out.printf(format:"%s%nTiempo universal: %s%nTiempo estandar: %s%n",
33            encabezado, t.aStringUniversal(), t.toString());
34    }
35 }
```

Clase PruebaAccesoMiembros:

```
J PruebaAccesoMiembros.java > PruebaAccesoMiembros > main(String[])
1
2 //Esta clase demuestra que Los miembros privados de Tiempo1 no son accesibles directamente.
3 //El código no compilará debido a las violaciones de encapsulamiento.
4
5 public class PruebaAccesoMiembros {
6     Run | Debug
7     public static void main(String[] args) {
8         Tiempo1 tiempo = new Tiempo1();
9         //Estas líneas causarán errores de compilación porque los miembros son privados
10        //Esto debe ser arreglado en un futuro
11        tiempo.hora = 7; // Error. Hora tiene acceso privado en Tiempo1
12        tiempo.minuto = 15; // Error. Minuto tiene acceso privado en Tiempo1
13        tiempo.segundo = 30; // Error. Segundo tiene acceso privado en Tiempo1
14    }
```

Clase PruebaThis

```
J PruebaThis.java >  TiempoSimple >  aStringUniversal()
1 //Clase de prueba para demostrar el uso de la referencia 'this'
2
3 public class PruebaThis {
4     Run | Debug
5     public static void main(String[] args) {
6         //Crear un objeto TiempoSimple
7         TiempoSimple tiempo = new TiempoSimple(hora:15, minuto:30, segundo:19);
8         //Mostrar el String creado que demuestra el uso de "this"
9         System.err.println(tiempo.crearString());
10    }
11 }
12
13 //Clase que demuestra el uso de la referencia 'this' para acceder a miembros de la clase
14
15 class TiempoSimple {
16     private int hora; //Hora del día
17     private int minuto; //Minuto de la hora
18     private int segundo; //Segundo del minuto
19
20
21     //Constructor que inicializa TiempoSimple
22     //Hora valor de la hora (0-23)
23     //Minuto valor del minuto (0-59)
24     //Segundo valor del segundo (0-59)
25
26     public TiempoSimple(int hora, int minuto, int segundo) {
27         //El uso explícito de "this" para distinguir variables de instancia de parámetros
28         this.hora = hora;
29         this.minuto = minuto;
30         this.segundo = segundo;
31     }
32
33
34     //Crea un String que muestra el uso de 'this'
35     //String que compara las llamadas con y sin 'this'
36
37     public String crearString() {
38         return String.format(format:"%24s: %s%n%24s: %s",
39             ...args:"this.aStringUniversal()", this.aStringUniversal(),
40             "aStringUniversal()", aStringUniversal());
41     }
42
43
44     //Devuelve la hora en formato universal de 24 Horas
45     //String con formato HH:MM:SS
46
47     public String aStringUniversal() {
48         //El uso de 'this' para acceder a miembros (Opcional en este momento)
49         return String.format(format:"%02d:%02d:%02d",
50             this.hora, this.minuto, this.segundo);
51     }
52 }
```