Directivas para las clases entidades

Construir las **clases entidades** siguiendo los siguientes criterios:

- Cada proyecto se llamará como la clase entidad a desarrollar, añadiendo el prefijo **Test**. Por ejemplo si la clase entidad se llama **Circulo** el proyecto se llamará **TestCirculo**.
- La clase con el punto de entrada al programa se llamará igual que el proyecto y pertenecerán al paquete **org.japo.java.basics.samples.**
- Las clases entidades pertenecerán al paquete **org.japo.java.basics.entities.**
- Los interfaces que eventualmente se definan pertenecerán al paquete **org.japo.java.basics.interfaces.**
- Cada clase entidad se denominará según el nombre asignado en su enunciado.
- Las clases entidades dispondrán del constructor predeterminado y al menos de un **constructor adicional** más que **inicialice todos los atributos** de la clase a los valores especificados en los parámetros y **verifique la corrección de los valores**.
- Los **atributos no constantes de la clase entidad** estarán **encapsulados** convenientemente y accedidos mediante sus **getters/setters públicos** respectivos.
- Como regla general, si el valor del **argumento** aplicado al setter **no es correcto** no se aplicará dicho valor.
- Los **atributos públicos y constantes** de las clases entidades serán marcados **estáticos**.
- Los métodos **públicos** de la clase entidad que **no utilicen ningún recurso de instancia de su propia clase** serán marcados como **estáticos**.
- Los atributos que sean públicos, constantes y estáticos, y los métodos públicos no estáticos de la clase, diferentes de constructores, métodos accesores y métodos sobrescritos, estarán definidos en un interfaz implementado por la clase entidad que se llamará igual que ésta pero con el prefijo I mayúscula. Por ejemplo, la clase **Circulo** tendrá el interfaz **ICirculo**.
- El programa de cada proyecto mostrará el funcionamiento de **todos los recursos** definidos en su respectiva entidad.

Definición de entidades

1. La clase entidad **Decimo** encapsula los datos de un décimo o billete de Lotería Nacional y su interfaz asociado es **IDecimo**. A los efectos de este ejercicio interesa manejar la siguiente información.
   a. El **número** del billete que siempre tiene cinco dígitos y va desde el número 00000 hasta el número 99999.
   b. La **serie** del número que va desde la serie 1 hasta la serie 160.
   c. La **fracción** del número llamada billete. Cada número se divide en diez fracciones, por eso también se llama décimo. Los décimos se numeran desde el 1 hasta el 10.

El constructor predeterminado generará los datos del billete de forma aleatoria, llamando al método:

```
private void generarDecimoAleatorio()
```

de tal forma, que el billete sea válido.

El constructor parametrizado comprobará que los datos pasados sean válidos. Si **algún dato** no fuese válido entonces **todos los datos** se generarán de forma aleatoria como en el constructor predeterminado.

Se encapsularán los campos variables de la clase **Decimo** teniendo en cuenta la validez de los valores de los métodos mutadores (**setters**). Si un valor no es correcto se desecha.

Además, la clase debe redefinir el método:

```
public String toString();
```

que al ser llamado genere un texto con los datos del billete según el siguiente formato

```
NNNNN – SSS - FF
```

En el billete, para el número tienen que aparecer los 5 dígitos, pero la serie y la fracción se justificarán con espacios.

También se debe redefinir el método:

```
public boolean equals(Object o);
```

que al ser llamado compare los datos del décimo actual con los datos del décimo pasado y devuelva **true** si todos los datos son iguales y **false** en caso contrario.

Adicionalmente, la clase **Decimo** debe definir el método:

```
public static boolean validarDecimo(int num, int ser, int fra);
```

que al ser llamado analiza la corrección de los valores de los argumentos, devolviendo **true** si un décimo con esos valores es válido y **false** si no lo es.

Por otro lado, la clase **Decimo** implementa el interfaz **IDecimo** que declara el método:

```
public void mostrarDecimo();
```

que al ser llamado muestra los datos del décimo actual ayudándose del método **toString**. El interfaz debe utilizarse para **declarar las constantes** que necesite la clase **Decimo**.

Por ultimo, codificar la clase **TestDecimo** que disponga de un **punto de entrada** para ejecutar el programa. Se debe instanciar un décimo utilizando el **constructor predeterminado** y otro décimo utilizando el **constructor parametrizado**. El programa debe mostrar por pantalla los datos de ambos décimos utilizando el método **mostrarDecimo** y debe compararlos utilizando el método **equals** e informar si son iguales o no.