

Ejercicios Prácticos

EE Programación

1. Diseñar en UML y codificar una clase Cancion

- a. Con los siguientes atributos:
 - i. título: variable cadena que guarde el título de una canción
 - ii. autor: variable cadena que guarde el autor de la canción
 - iii. duración: variable entera indicando los segundos de la canción
- b. Con los siguientes constructores y métodos:
 - i. Cancion() constructor por default
 - ii. Cancion(String, String) constructor que recibe el título y autor de la canción
 - iii. Cancion(String, String, int) constructor que recibe el título, autor y segundos de duración de una canción
 - iv. getTitulo(): regresa el título de la canción
 - v. getAutor(): regresa el autor de la canción
 - vi. getDuracion(): regresa la duración de la canción
 - vii. setTitulo(String): establece el título de la canción
 - viii. setAutor(String): establece el título de la canción
 - ix. setDuracion(int): establece la duración de la canción

Considerando la clase Canción, es necesario la existencia de una clase adicional para poder generar Álbumes, modifica tu diagrama UML para tener la clase Álbum, la cual debería permitir tener múltiples canciones (de más de 1) y asignar un artista o interprete, nombre y año al álbum. Es necesario implementar esta clase e incluir los siguientes métodos: agregar(Cancion), eliminar(int), getCancion(int) que regresa la canción dado un índice, y getNumeroDeCanciones() que regresa el total de canciones agregadas al álbum.

Una vez implementadas las dos clases será necesario probarlas con la siguiente clase de prueba:

```
//Clase TestAlbum
public class TestAlbum{
    public static void main(String[] args){
        System.out.println("Probando clase Cancion y Album");
        Cancion c0 = new Cancion();
        c0.setAutor("The Black Eyed Peas");
        c0.setTitulo("Boom Boom Pow");
        c0.setDuracion(260);
        Cancion c1 = new Cancion("The Black Eyed Peas", "Rock That Body", 289);

        Cancion c2 = new Cancion("The Black Eyed Peas", "Meet Me Halfway");
        c2.setDuracion(277);

        Album a1 = new Album("The E.N.D.", "The Black Eyed Peas", 2009);
        a1.agregarCancion(c0);
        a1.agregarCancion(c1);
        a1.agregarCancion(c2);
        a1.agregarCancion(new Cancion("The Black Eyed Peas", "Imma Be", 302));

        System.out.println("Mostrando información del Álbum: " + a1.getNombre());
        System.out.println("Interprete: " + a1.getInterprete());
        System.out.println("Año: " + a1.getAnio());
        System.out.println("\nCanciones (" + a1.getNumeroCanciones() + "): ");
        for (int i = 0; i < a1.getNumeroCanciones(); i++){
            System.out.println((i+1) + ". " + a1.getCancion(i).getAutor() + " - " +
a1.getCancion(i).getTitulo() + "\t" + a1.getCancion(i).getDuracion() + " segs.");
        }
    }
}
```

La salida esperada después de ejecutar TestAlbum es la siguiente:

```
Probando clase Cancion y Album
Mostrando información de Álbum: The E.N.D.
Interprete: The Black Eyed Peas
Año: 2009
```

```
Canciones (4):
1. The Black Eyed Peas - Boom Boom Pow      260 segs.
2. The Black Eyed Peas - Rock That Body      289 segs.
3. The Black Eyed Peas - Meet Me Halfway     277 segs.
4. The Black Eyed Peas - Imma Be             302 segs.
```

2. Diseñar y codificar una clase llamada EstadisticaDescriptiva que permita construir una instancia que rellene un arreglo de longitud 10 y de tipo entero con números aleatorios. La generación de números aleatorios y relleno del arreglo debe de estar entre un límite superior y un límite inferior proporcionado en el constructor. Después de crear la instancia será posible invocar: un método llamado promediar() que haga las operaciones necesarias para obtener la media, un método sumar() para realizar la suma de todos los elementos del arreglo, y un método llamado desviacionEstandar(), el cual calcula la desviación estándar.

Para la generación de números aleatorios utilizar la función Math.random() que devuelve un número real aleatorio entre 0 y 1.

Para generar un número aleatorio entre un límite superior e inferior podría utilizar el siguiente código:

```
int limite_superior = 10;
int limite_inferior = 5;

int na = (int) ((Math.random() * (limite_superior - limite_inferior)) + limite_inferior);
```

Crear una clase de prueba que muestre lo siguiente (los números del arreglo no tienen que ser iguales a los que se enlistan a continuación).

Nota: La impresión de los valores únicamente debe estar en la clase de prueba EstadisticaDescriptivaTest

Clase EstadisticaDescriptivaTest para probar la clase EstadisticaDescriptiva
Números generados en el arreglo: 5, 10, 9, 6, 7, 8, 9, 5, 7, 6
Suma: 72
Media: 7.2
Desviación Estándar: 1.75

La función para calcular la media y desviación estándar son las siguientes:

$$media = \frac{\sum_{i=1}^n x_i}{n} \quad desviación = \sqrt{\frac{\sum_{i=1}^n (x_i - media)^2}{n - 1}}$$

3. Escribe un programa que pida diez números enteros y los muestre en orden inverso (del último leído hasta el primero).
4. Crea las siguientes clases (cada una en su archivo):
- Motor: con métodos para arrancar el motor y apagarlo.
 - Rueda: con métodos para inflar la rueda y desinflarla.
 - Ventana: con métodos para abrirla y cerrarla.
 - Puerta: con una ventana y métodos para abrir la puerta y cerrar la puerta.
 - Coche: con un motor, cuatro ruedas y dos puertas; con los métodos que te parezcan adecuados
 - Implementar la clase de prueba que muestre que las clases codificadas funcionan correctamente.
5. Crea una clase Cuenta (bancaria) con atributos para el número de cuenta (un entero largo), el DNI del cliente (otro entero largo), el saldo actual y el interés anual que se aplica a la cuenta (porcentaje). Define en la clase los siguientes métodos:
- Constructor por defecto y constructor con DNI, saldo e interés
 - Getters y setters. Para el número de cuenta no habrá set.
 - actualizarSaldo(): actualizará el saldo de la cuenta aplicándole el interés diario (interés anual dividido entre 365 aplicado al saldo actual).
 - ingresar(double): permitirá ingresar una cantidad en la cuenta.
 - retirar(double): permitirá sacar una cantidad de la cuenta (si hay saldo).
 - Método que nos permita mostrar todos los datos de la cuenta.
 - El número de cuenta se asignará de forma correlativa a partir de 100001, asignando el siguiente número al último asignado.
 - Implementar la clase de prueba que muestre que las clases codificadas funcionan correctamente.