Práctica N°7 de SIS103

Nombre: Gonzales Suyo Franz Reinaldo **C.U.** 111-500

Carrera: Ing. en Ciencias de la Computación

Asignación de objetos:

Clase Avión:

```
/**
 * Esta clase define objetos de tipo Avion con un fabricante y número de
 * motores como atributos.
 *
 * @version 1.2/2020
 */
public class Avion {
    private String fabricante; /* Atributo que define el nombre del fabri cante
del avion */
    private int numeroMotores; /* Atributo que define el número de mo tores del
avion */

    /**
     * Constructor de la clase Avion
     *
     * @param fabricante    Parámetro que define el nombre del fabricante
     *                      de un avion
     * @param numeroMotores Parámetro que define el número de
     *                      motores que tiene un avion
     */
    private Avion(String fabricante, int numeroMotores) {
        this.fabricante = fabricante;
        this.numeroMotores = numeroMotores;
    }

    /**
     * Método que devuelve el nombre del fabricante de un avion
     *
     * @return El nombre del fabricante de un avion
     */
    public String getFabricante() {
        return fabricante;
    }

    /**
     * Método que establece el nombre de un fabricande de un avion
     *
     * @param fabricante Parámetro que define el nombre del fabricante
     *                   de un avion
     */
    private void setFabricante(String fabricante) {
        this.fabricante = fabricante;
    }
```

```java
/**
* Método que cambia el fabricante de un avion pasado como
* parámetro por el valor “Lockheed”
* @param a Parámetro que define un avion
*/
private void cambiarFabricante(Avion a) {
    a.setFabricante("Lockheed");
}

/**
 * Método que devuelve el número de motores de un avion
 *
 * @return El número de motores de un avion
 */
public int getNumeroMotores() {
    return numeroMotores;
}

/**
 * Método que establece el número de motores de un avion
 *
 * @param numeroMotores Parámetro que define el número de
 *                      motores de un avion
 */
private void setNumeroMotores(int numeroMotores) {
    this.numeroMotores = numeroMotores;
}

/**
* Método que imprime en pantalla el fabricante de un avion
*/
public void imprimirFabricante() {
    System.out.println("El fabricante del avion es: " + fabricante);
}

/**
* Método main que crea dos aviones y modifica sus fabricantes
*/
public static void main(String args[]) {
    Avion a1 = new Avion("Airbu", 4);
    Avion a2;
    Avion a3 = new Avion("Boeing", 4);
    a2 = a1;

    Avion a4 = new Avion("Aerosur", 5);
    Avion a5;
    a5 = a4;

    a1.imprimirFabricante();
    a2.imprimirFabricante();
    a1.setFabricante("Douglas");
    a1.imprimirFabricante();
```

```
        a2.imprimirFabricante();
        a1.cambiarFabricante(a2);
        a2.imprimirFabricante();

        a4.imprimirFabricante();
        a5.imprimirFabricante();

        a5.setFabricante("Stealth");
        a4.imprimirFabricante();
    }
}
```

Prueba de Ejecución:

```
PS C:\GONZALES\SIS103> cd C:\GONZALES\SIS103\Asignacion_Objetos
PS C:\GONZALES\SIS103\Asignacion_Objetos> javac Avion.java
PS C:\GONZALES\SIS103\Asignacion_Objetos> java Avion
El fabricante del avion es: Airbu
El fabricante del avion es: Airbu
El fabricante del avion es: Douglas
El fabricante del avion es: Douglas
El fabricante del avion es: Lockheed
El fabricante del avion es: Aerosur
El fabricante del avion es: Aerosur
El fabricante del avion es: Stealth
PS C:\GONZALES\SIS103\Asignacion_Objetos>
```

Clase ConversorMetros:

```
/**
 * Esta clase define objetos de tipo ConversorMetros el cual permite
 * realizar conversiones entre diferentes unidades de medición de longitud.
 *
 * @version 1.2/2020
 */
public class ConversorMetros {
    /*
     * Atributo que define la cantidad de metros a convertir a diferentes
     * unidades de longitud
     */
    double metros;
    final int FACTOR_METROS_CM = 100; /*
                                       * Factor de conversión de
```

```
                                        * metros a centimetros
                                        */
    final int FACTOR_METROS_MILIM = 1000; /*
                                           * Factor de conversión
                                           * de metros a milimetros
                                           */
    final double FACTOR_METROS_PULGADAS = 39.37; /*
                                                  * Factor de
                                                  * conversión de metros a
pulgadas
                                                  */
    final double FACTOR_METROS_PIES = 3.28; /*
                                             * Factor de
                                             * conversión de metros a pies
                                             */
    final double FACTOR_METROS_YARDAS = 1.09361; /*
                                                  * Factor de
                                                  * conversión de metros a
yardas
                                                  */

    /**
     * Constructor de la clase ConversorMetros
     * @param metros Parámetro que define la cantidad de metros a
     *               convertir a otras unidades de longitud
     */
    public ConversorMetros(double metros) {
        this.metros = metros;
    }

    /**
     * Método que convierte metros a centimetros
     *
     * @return Resultado de la conversión de metros a centimetros
     */
    public double convertirMetrosToCentimetros() {
        double centimetros;
        centimetros = FACTOR_METROS_CM * metros;
        return centimetros;
    }

    /**
     * Método que convierte metros a milimetros
     *
     * @return Resultado de la conversión de metros a milimetros
     */
    public double convertirMetrosToMilímetros() {
        double milimetros;
        milimetros = FACTOR_METROS_MILIM * metros;
        return milimetros;
    }

    /**
```

```java
     * Método que convierte metros a pulgadas
     *
     * @return Resultado de la conversión de metros a pulgadas
     */
    public double convertirMetrosToPulgadas() {
        double pulgadas;
        pulgadas = FACTOR_METROS_PULGADAS * metros;
        return pulgadas;
    }

    /**
     * Método que convierte metros a pies
     *
     * @return Resultado de la conversión de metros a pies
     */
    public double convertirMetrosToPies() {
        double pies;
        pies = FACTOR_METROS_PIES * metros;
        return pies;
    }

    /**
     * Método que convierte metros a yardas
     *
     * @return Resultado de la conversión de metros a yardas
     */
    public double convertirMetrosToYardas() {
        double yardas;
        yardas = FACTOR_METROS_YARDAS * metros;
        return yardas;
    }

    /**
* Método main que define una cierta cantidad de metros y los
* convierte a diferentes unidades de longitud
*/
    public static void main (String args[]) {

        ConversorMetros conversor = new ConversorMetros(3.5);

        System.out.println("Metros = " + conversor.metros);
        System.out.println("Metros a centimetros = " +
conversor.convertirMetrosToCentimetros());
        System.out.println("Metros a milimetros = " +
conversor.convertirMetrosToMilímetros());
        System.out.println("Metros a pulgadas = " +
conversor.convertirMetrosToPulgadas());
        System.out.println("Metros a pies = " +
conversor.convertirMetrosToPies());
        System.out.println("Metros a yardas = " +
conversor.convertirMetrosToYardas());
    }
}
```

Prueba de ejecución:

```
PS C:\GONZALES\SIS103\Asignacion_Objetos> javac ConversorMetros.java
PS C:\GONZALES\SIS103\Asignacion_Objetos> java ConversorMetros
Metros = 3.5
Metros a centimetros = 350.0
Metros a milimetros = 3500.0
Metros a pulgadas = 137.795
Metros a pies = 11.479999999999999
Metros a yardas = 3.827635
PS C:\GONZALES\SIS103\Asignacion_Objetos>
```

Ejercicios Propuestos:

1. Clase ConversorSuperficie:

```
2.
3. public class ConversorSuperficie {
4.
5.     // Inicilizamos las variables de medidas
6.     final int FACTOR_METROS_AREA = 100;
7.     final int FACTOR_METROS_HECTARIA = 10000;
8.     final int FACTOR_METROS_KILOMETRO_CUADRADO = 1000000;
9.     final int FACTOR_METROS_FANEGA = 6460;
10.    final double FACTOR_METROS_ACRE = 4046.85;
11.
12.    public int metro_cuadrado;
13.
14.    public ConversorSuperficie(int metro_cuadrado){
15.        this.metro_cuadrado = metro_cuadrado;
16.    }
17.
18.    // función para obtener el calculo de metros cuadrados a su area
19.    public double convertirToArea(){
20.        double area = metro_cuadrado / FACTOR_METROS_AREA;
21.        return area;
22.    }
23.
24.    // Método para obtener el calculo de hectareas a su superficie en
   metros cuadrados
25.    public double convertirToHectareas(){
26.        double hectareas = metro_cuadrado / FACTOR_METROS_HECTARIA;
27.        return hectareas;
28.    }
29.
30.    // Método para obtener el calculo de kilómetros cuadrados a su
   superficie en metros cuadrados
31.    public double convertirToKilometrosCuadrados(){
```

```java
        double kilometros_cuadrados = metro_cuadrado / FACTOR_METROS_KILOMETRO_CUADRADO;
        return kilometros_cuadrados;
    }

    // Método para obtener el calculo de fanegas a su superficie en metros cuadrados
    public double convertirToFanegas(){
        double fanegas = metro_cuadrado / FACTOR_METROS_FANEGA;
        return fanegas;
    }

    // Método para obtener el calculo de acres a su superficie en metros cuadrados
    public double convertirToAcres(){
        double acres = metro_cuadrado / FACTOR_METROS_ACRE;
        return acres;
    }

    public static void main(String[] args) {

        ConversorSuperficie conversor = new ConversorSuperficie(20000);

        System.out.println("Superficie = " + conversor.convertirToArea() + " m²");
        System.out.println("Superficie = " + conversor.convertirToHectareas() + " ha");
        System.out.println("Superficie = " + conversor.convertirToKilometrosCuadrados() + " km²");
        System.out.println("Superficie = " + conversor.convertirToFanegas() + " fanegas");
        System.out.println("Superficie = " + conversor.convertirToAcres() + " acres");
    }

}

```

Prueba de ejecución:

```
PS C:\GONZALES\SIS103\Asignacion_Objetos> javac ConversorSuperficie.java
PS C:\GONZALES\SIS103\Asignacion_Objetos> java ConversorSuperficie
Superficie = 200.0 m²
Superficie = 2.0 ha
Superficie = 0.0 km²
Superficie = 3.0 fanegas
Superficie = 4.942115472528016 acres
PS C:\GONZALES\SIS103\Asignacion_Objetos>
```

2. Clase ConversorVolumen:

```
public class ConversorVolumen {

    public int litros;

    final double FACTOR_LITROS_GALON = 4.41;
    final double FACTOR_LITROS_PINTA = 0.46;
    final double FACTOR_LITROS_BARRIL = 158.99;
    final int FACTOR_LITROS_METROS_CUBICO = 1000;
    final int FACTOR_LITROS_HECTOLITRO = 100;


    public ConversorVolumen(int litros) {
        this.litros = litros;
    }

    // Convertir litros a galones
    public double convertirToGalones(){
        double galones = litros / FACTOR_LITROS_GALON;
        return galones;
    }

    // Convertir litros a pintas
    public double convertirToPintas(){
        double pintas = litros / FACTOR_LITROS_PINTA;
        return pintas;
    }

    // Convertir litros a barriles
    public double convertirToBarriles(){
        double barriles = litros / FACTOR_LITROS_BARRIL;
        return barriles;
    }

    // Convertir litros a metros cubicos
    public double convertirToMetrosCubicos(){
        double metroCubico = litros / FACTOR_LITROS_METROS_CUBICO;
        return metroCubico;
    }

    // Convertir litros a hectolitros
    public double convertirToHectolitros(){
        double hectolitros = litros / FACTOR_LITROS_HECTOLITRO;
        return hectolitros;
    }

    public static void main(String[] args) {

        ConversorVolumen conversorVolumen = new ConversorVolumen(100);

        System.out.println("Litros en galones: " +
conversorVolumen.convertirToGalones());
```

```
        System.out.println("Litros en pintas: " +
conversorVolumen.convertirToPintas());
        System.out.println("Litros en barriles: " +
conversorVolumen.convertirToBarriles());
        System.out.println("Litros en metros cubicos: " +
conversorVolumen.convertirToMetrosCubicos());
        System.out.println("Litros en hectolitros: " +
conversorVolumen.convertirToHectolitros());
    }
}
```

Prueba de ejecución:

```
PS C:\GONZALES\SIS103\Asignacion_Objetos> javac ConversorVolumen.java
PS C:\GONZALES\SIS103\Asignacion_Objetos> java ConversorVolumen
Litros en galones: 22.675736961451246
Litros en pintas: 217.39130434782606
Litros en barriles: 0.6289703754953141
Litros en metros cubicos: 0.0
Litros en hectolitros: 1.0
PS C:\GONZALES\SIS103\Asignacion_Objetos>
```