

Tema 7 Actividad 2

Índice

Ejercicio7_1_1.

Crea una clase Person, que debe comportarse de la siguiente manera:

```
1 Person ada = new Person("Ada Lovelace", "24 Maddox St. London W1S 2QN");
2 Person javier = new Person("Javier García", "Calle Mayor 15 12002 Castellón");
3 System.out.println(ada);
4 System.out.println(javier);
```

```
Ada Lovelace
24 Maddox St. London W1S 2QN
Javier García
Calle Mayor 15 12002 Castellón
```

```
package ejercicio7_2_1;
public class Ejercicio7_2_1
{
    public static void main(String[] args)
    {
        Person ada = new Person("Ada Lovelace", "24 Maddox St. London W1S 2QN");
        Person javier = new Person("Javier García", "Calle Mayor 15 12002 Castellón");
        System.out.println(ada);
        System.out.println(javier);
    }

    public class Person
    {
        private String nombre;
        private String direccion;

        public Person(String nombre, String direccion)
        {
            this.nombre = nombre;
            this.direccion = direccion;
        }

        @Override
        public String toString()
        {
            return nombre + "" + "\n" + direccion;
        }
    }
}
```

```

run:
Ada Lovelance
 24 Maddox St. London W1S 2QN
Javier García
  Calle Mayor 15 12002 Castellón
BUILD SUCCESSFUL (total time: 0 seconds)
|

```

Ejercicio7_1_2.

Implementa una clase para gestionar un aparato de aire acondicionado con los atributos: nombre, t. mínima, t. máxima y t. actual.

1. Constructor: Recibe nombre, min y max. La temperatura actual se inicializa automáticamente en la máxima.
2. Métodos: subir() y bajar() modifican la temperatura grado a grado, impidiendo superar los límites establecidos.
3. toString(): Devuelve el estado. Debe añadir texto informativo según la temperatura:
 - Si está al mínimo: " MÁXIMO FRÍO"
 - Si está al máximo: " MODO ECO"
 - Caso contrario: " Normal"

```

public class AireAcondicionado
{
    private String nombre;
    private int t_Minima;
    private int t_Máxima;
    private int t_Actual;

    public AireAcondicionado(String nombre, int min, int max)
    {
        this.nombre = nombre;
        this.t_Minima = min;
        this.t_Máxima = max;
        this.t_Actual = max;
    }

    public void subir()
    {
        if(t_Actual != t_Máxima)
            t_Actual++;
    }

    public void bajar()
    {
        if(t_Actual != t_Minima)
            t_Actual--;
    }

    @Override
    public String toString()
    {
        String texto = nombre + "[" + t_Actual + "°C] ";
        if(t_Actual == t_Máxima)
            texto += "MODO ECO";
        else if(t_Actual == t_Minima)
            texto += "MÁXIMO FRÍO";
        else
            texto += "Normal";
        return texto;
    }
}

```

```

public class Ejercicio7_2_2
{
    public static void main(String[] args)
    {
        AireAcondicionado Aire_Samsung = new AireAcondicionado ("Aire Samsung", 18, 22);

        System.out.println("-- ESTADO INICIAL ---");
        System.out.println(Aire_Samsung);
        System.out.println("");

        System.out.println("-- BAJAMOS TEMPERATURA ---");
        Aire_Samsung.bajar();
        System.out.println(Aire_Samsung);
        Aire_Samsung.bajar();
        System.out.println(Aire_Samsung);
        System.out.println("");

        System.out.println("-- PROBAMOS EL MÍNIMO ---");
        Aire_Samsung.bajar();
        Aire_Samsung.bajar();
        Aire_Samsung.bajar();
        Aire_Samsung.bajar();
        Aire_Samsung.bajar();
        Aire_Samsung.bajar();
        Aire_Samsung.bajar();
        System.out.println(Aire_Samsung);
    }
}

```

```

run:
-- ESTADO INICIAL ---
Aire Samsung[22°C] MODO ECO

-- BAJAMOS TEMPERATURA ---
Aire Samsung[21°C] Normal
Aire Samsung[20°C] Normal

-- PROBAMOS EL MÍNIMO ---
Aire Samsung[18°C] MÁXIMO FRÍO
BUILD SUCCESSFUL (total time: 0 seconds)

```

Ejercicio7_1_3.

Implementa la clase Cuadrado, que representa esta figura geométrica mediante dos puntos en un plano donde la Y crece hacia abajo (coordenadas de pantalla).

1. Atributos: La clase debe almacenar 4 números decimales (double) que representan dos esquinas opuestas: • (x1, y1): Esquina superior izquierda. • (x2, y2): Esquina inferior derecha.
2. Constructor: Debe recibir los 4 valores en este orden: (x1, y1, x2, y2).
3. Métodos de Cálculo: • Perimetro(): Devuelve la suma de los cuatro lados. • Area(): Devuelve la superficie del cuadrado • Diagonal(): Devuelve la distancia entre esquinas opuestas.
4. toString(): Se encargará de mostrar la información del cuadrado, tal y como se muestra en el ejemplo de salida.
5. Se han de crear los getters y los setters.
6. Modificar con los setters los valores iniciales y volver a calcular el área, perímetro y diagonal

Ejemplo de salida:

```
==== ESTADO INICIAL ====
Cuadrado definido por (2.0, 2.0) y (6.0, 6.0)
Área:      16.0
Perímetro: 16.0
Diagonal:  5.66

... Modificando coordenadas con Setters (Haciéndolo más grande) ...

==== ESTADO FINAL ====
Cuadrado definido por (2.0, 2.0) y (12.0, 12.0)
Área:      100.0
Perímetro: 40.0
Diagonal: 14.14
```

```
{  
    private double x1;  
    private double x2;  
    private double y1;  
    private double y2;  
  
    public Cuadrado(double x1, double y1, double x2, double y2)  
    {  
        this.x1 = x1;  
        this.x2 = x2;  
        this.y1 = y1;  
        this.y2 = y2;  
    }  
  
    public double perimetro()  
    {  
        return Math.abs((x2 - x1) * 4);  
    }  
  
    public double area()  
    {  
        return Math.pow((x2-x1), 2);  
    }  
  
    public double diagonal()  
    {  
        return Math.sqrt((Math.pow((x1 - x2), 2))+(Math.pow((y1 - y2), 2)));  
    }  
  
    public void setCuadrado(double x1, double y1, double x2, double y2)  
    {  
        this.x1 = x1;  
        this.x2 = x2;  
        this.y1 = y1;  
        this.y2 = y2;  
    }  
  
    @Override  
    public String toString()  
    {  
        return "Cuadrado definido por (" + x1 + ", " + y1 + ") y (" + x2 + ", " + y2 + ")";  
    }  
}
```

```

package ejercicio7_2_3;
public class Ejercicio7_2_3
{
    public static void main(String[] args)
    {
        Cuadrado cuadrado = new Cuadrado(2.0, 2.0, 6.0, 6.0);

        System.out.println("==== ESTADO INICIAL ====");
        System.out.println(cuadrado.toString());
        System.out.printf("%-10s %.2f %n", "Área:", cuadrado.area());
        System.out.printf("%-10s %.2f %n", "Perímetro:", cuadrado.perimetro());
        System.out.printf("%-10s %.2f %n%n", "Diagonal:", cuadrado.diagonal());

        System.out.println("... Modificando coordenadas con Setters (Haciéndolo más grande) ...");

        cuadrado.setCuadrado(2.0, 2.0, 12.0, 12.0);
        System.out.println("");
        System.out.println("==== ESTADO FINAL ====");
        System.out.println(cuadrado.toString());
        System.out.printf("%-10s %.2f %n", "Área:", cuadrado.area());
        System.out.printf("%-10s %.2f %n", "Perímetro:", cuadrado.perimetro());
        System.out.printf("%-10s %.2f %n", "Diagonal:", cuadrado.diagonal());
    }
}

```

```

==== ESTADO INICIAL ====
Cuadrado definido por (2.0, 2.0) y (6.0, 6.0)
Área: 16,00
Perímetro: 16,00
Diagonal: 5,66

... Modificando coordenadas con Setters (Haciéndolo más grande) ...

==== ESTADO FINAL ====
Cuadrado definido por (2.0, 2.0) y (12.0, 12.0)
Área: 100,00
Perímetro: 40,00
Diagonal: 14,14
BUILD SUCCESSFUL (total time: 0 seconds)
|
```

Ejercicio 7_1_4.

Crea un programa con una clase llamada Persona con los siguientes atributos:

Atributos: dni, nombre, apellidos y edad. Realiza las siguientes acciones:

- En el main de la clase principal crea dos objetos de la clase Persona.

- En este caso el constructor estará vacío y no dará valores a los atributos.

Luego, pide por teclado los datos de ambas personas (guárdalos en los objetos). Para ello haz uso de los setters.

Por último, y haciendo uso del método `toString()` imprime dos mensajes por pantalla (uno por objeto) con un mensaje del estilo “Azucena Luján García con DNI ... es / no es mayor de edad”.

Ejemplo de salida:

```
--- Introduce datos para la Persona 1 ---
Introduce el Nombre: Azucena
Introduce los Apellidos: Luján García
Introduce el DNI: 12345678Z
Introduce la Edad: 25

--- Introduce datos para la Persona 2 ---
Introduce el Nombre: Pedro
Introduce los Apellidos: Ramirez
Introduce el DNI: 87654321X
Introduce la Edad: 15

--- RESULTADOS ---
Azucena Luján García con DNI 12345678Z es mayor de edad
Pedro Ramirez con DNI 87654321X no es mayor de edad
```

```
    private String dni;
    private String nombre;
    private String apellidos;
    private int edad;

    public Persona()
    {

    }

    public void setDni(String dni) {
        this.dni = dni;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public void setApellidos(String apellidos) {
        this.apellidos = apellidos;
    }

    public void setEdad(int edad) {
        this.edad = edad;
    }

    @Override
    public String toString()
    {
        String texto = nombre + " " + apellidos + " con DNI " + dni;

        if(edad >= 18)
            texto += " es mayor de edad";
        else
            texto += " no es mayor de edad";

        return texto;
    }
}
```

```
public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    Persona personal = new Persona();
    Persona persona2 = new Persona();

    String respuesta = "";
    int edad = 0;

    System.out.println("--- Introduce datos para la Persona 1 ---");
    System.out.print("Introduce el Nombre: ");
    respuesta = sc.nextLine();
    personal.setNombre(respuesta);
    System.out.print("Introduce los Apellidos: ");
    respuesta = sc.nextLine();
    personal.setApellidos(respuesta);
    System.out.print("Introduce el DNI: ");
    respuesta = sc.nextLine();
    personal.setDni(respuesta);
    System.out.print("Introduce la Edad: ");
    edad = sc.nextInt();
    personal.setEdad(edad);
    System.out.println("");
    sc.nextLine();
```

```
System.out.println("--- Introduce datos para la Persona 2 ---");
System.out.print("Introduce el Nombre: ");
respuesta = sc.nextLine();
personal2.setNombre(respuesta);
System.out.print("Introduce los Apellidos: ");
respuesta = sc.nextLine();
personal2.setApellidos(respuesta);
System.out.print("Introduce el DNI: ");
respuesta = sc.nextLine();
personal2.setDni(respuesta);
System.out.print("Introduce la Edad: ");
edad = sc.nextInt();
personal2.setEdad(edad);
System.out.println("");

System.out.println("--- RESULTADOS ---");
System.out.println(personal.toString());
System.out.println(persona2.toString());
```

```

run:
--- Introduce datos para la Persona 1 ---
Introduce el Nombre: Azucena
Introduce los Apellidos: Luján García
Introduce el DNI: 12345678Z
Introduce la Edad: 25

--- Introduce datos para la Persona 2 ---
Introduce el Nombre: Pedro
Introduce los Apellidos: Ramirez
Introduce el DNI: 87654321X
Introduce la Edad: 15

--- RESULTADOS ---
Azucena Luján García con DNI 12345678Z es mayor de edad
Pedro Ramirez con DNI 87654321X no es mayor de edad
BUILD SUCCESSFUL (total time: 39 seconds)

```

Ejercicio 7_1_5.

Implementa una clase llamada Alumno para gestionar las calificaciones de un estudiante, con las siguientes características:

1. Constructor: o Las notas se inicializarán automáticamente a 1.0.
2. Métodos: o `toString()`: Devuelve la información del alumno. Debe mostrar el nombre, la nota media y un mensaje final dependiendo del resultado: 1. Si la media es menor a 5: " Debe Recuperar" 2. Si la media está entre 5 y 8.9: " Aprobado" 3. Si la media es 9 o superior: " Sobresaliente"
3. En el main: o Instancia tres objetos Alumno pasando un nombre. o Asignale tres notas usando los setters. o Muestra la información final usando `toString()`.

Ejemplo de salida:

```

--- FICHA ALUMNO 1 ---
Alumno: Luis Torres
Nota Media: 3.33          // Notas seteadas: (3.5 + 4.0 + 2.5) / 3
Estado: ● Debe Recuperar // Sale esto porque 3.33 es menor que 5

--- FICHA ALUMNO 2 ---
Alumno: Ana Ruiz
Nota Media: 6.33          // Notas seteadas: (6.0 + 7.5 + 5.5) / 3
Estado: ✓ Aprobado      // Sale esto porque 6.33 está entre 5 y 9

--- FICHA ALUMNO 3 ---
Alumno: Sofía Mendiola
Nota Media: 9.50          // Notas seteadas: (10.0 + 9.5 + 9.0) / 3
Estado: ★ Sobresaliente // Sale esto porque 9.50 es mayor o igual a 9

```

```
private String nombre;
private double nota1;
private double nota2;
private double nota3;

public Alumno(String nombre)
{
    this.nombre = nombre;
    this.nota1 = 1.0;
    this.nota2 = 1.0;
    this.nota3 = 1.0;
}

public void setNota1(double nota1) {
    this.nota1 = nota1;
}

public void setNota2(double nota2) {
    this.nota2 = nota2;
}

public void setNota3(double nota3) {
    this.nota3 = nota3;
}

public double calcularMedia()
{
    return Math.round(((nota1 + nota2 + nota3) / 3)* 100.0) / 100.0;
}
```

```
@Override
public String toString()
{
    String texto = "Alumno: " + nombre + "\nNota Media: " + calcularMedia() + "\nEstado: ";

    if(calcularMedia() < 5)
        texto += "Debe Recuperar";
    else if(calcularMedia() >= 5 && calcularMedia() < 9)
        texto += "Aprobado";
    else
        texto += "Sobresaliente";
    return texto;
}
```

```

public static void main(String[] args)
{
    Alumno alumno1 = new Alumno("Luis Torres");
    Alumno alumno2 = new Alumno("Ana Ruiz");
    Alumno alumno3 = new Alumno("Sofia Mendiola");

    System.out.println("--- FICHA ALUMNO 1 ---");
    alumno1.setNota1(3.5);
    alumno1.setNota2(4.0);
    alumno1.setNota3(2.5);
    System.out.println(alumno1.toString());
    System.out.println("");

    System.out.println("--- FICHA ALUMNO 2 ---");
    alumno2.setNota1(6.0);
    alumno2.setNota2(7.5);
    alumno2.setNota3(5.5);
    System.out.println(alumno2.toString());
    System.out.println("");

    System.out.println("--- FICHA ALUMNO 3 ---");
    alumno3.setNota1(10.0);
    alumno3.setNota2(9.5);
    alumno3.setNota3(9.0);
    System.out.println(alumno3.toString());
    System.out.println("");
}

```

```

lauri.
--- FICHA ALUMNO 1 ---
Alumno: Luis Torres
Nota Media: 3.33
Estado: Debe Recuperar

--- FICHA ALUMNO 2 ---
Alumno: Ana Ruiz
Nota Media: 6.33
Estado: Aprobado

--- FICHA ALUMNO 3 ---
Alumno: Sofia Mendiola
Nota Media: 9.5
Estado: Sobresaliente

```