

Clean Code

En esta practica voy a enseñar capturas de un programa hecho en la asignatura de programación. Es un proyecto de java y su funcionalidad es la creación de puntos que tendrán un id y un contador de cuantos puntos hay. A su vez tenemos un método con el que dándole las coordenadas de tres puntos nos calcula el área del triangulo que crean entre los puntos

Hay que comentar que el código no ha sido la prioridad para hacer el trabajo, es decir, no me he centrado en generar un código completamente formal, ya que me he enfocado en hacer el código limpio y claro siguiendo los pasos del clean code. Ejemplo de esto es que los atributos de la clase Punto2D no están encapsulados pero debido a que eso no concierne ni interfiere en el objetivo de la practica lo he dejado así, también he recortado cosas como el constructor de la clase ya que por ejemplo este no puede cumplir que el nombre del método sea un atributo ya que tiene que ser el mismo nombre que la clase.

Captura de la clase Punto2D :

```
1 package daml_delafuentecedillo_jonathan_ej0501b10punto3d;
2
3 public class Punto2D {
4
5     public float X;
6     public float Y;
7     private static int contador = 0;
8     private static String punto = "Punto";
9     final private int id;
```

(Creación de la clase y atributos)

-El nombre de la clase es un nombre lo cual es bueno aun que este programa al solo contener puntos en un plano de dos dimensiones el nombre puede ser redundante e innecesario, por lo cual lo cambiare a "Punto".

Los nombres son pronunciables y con significado aun que podría darse el caso de gente que el nombre "X" e "Y" para las variables de las coordenadas X e Y del punto, por ende cambiare el nombre de la variable por "coordenadaX" y "coordenadaY".

-No hablare del segundo apartados de funciones en esta parte ya que carece de funciones.

-En cuanto a la tercera parte de comentarios no hace falta poner ninguno innecesariamente ya que los nombres de las variables son auto explicativas, podría hacer uno que explique que la variable "punto" es simplemente para nombrar el objeto pero no lo veo necesario ya que cualquiera podría entenderlo.

```

12 public static int getContador()
13     return contador;
14 }
15
16 public static String getPunto() {
17     return punto;
18 }
19
20 public double distancia2D (Punto2D Param){
21     double distancia_calculada = Math.sqrt(Math.pow(Param.X-X, b: 2)+Math.pow(Param.Y-Y, b: 2));
22     return distancia_calculada;
23 }
24
25 public String distancia2D (double ParamX1, double ParamY1, double ParamX2, double ParamY2, double ParamX3, double ParamY3){
26     double distanciaAB = Math.sqrt(Math.pow(ParamX1-ParamX2, b: 2)+Math.pow(ParamY1-ParamY2, b: 2));
27     double distanciaAC = Math.sqrt(Math.pow(ParamX1-ParamX3, b: 2)+Math.pow(ParamY1-ParamY3, b: 2));
28     double distanciaBC = Math.sqrt(Math.pow(ParamX2-ParamX3, b: 2)+Math.pow(ParamY2-ParamY3, b: 2));
29     char SimboloElevado = 94;
30     double semiperimetro = (distanciaAB + distanciaAC + distanciaBC)/2;
31     double areaPrimeraForma = Math.sqrt(semiperimetro * (semiperimetro-distanciaAB) * (semiperimetro-distanciaAC) * (semiperimetro-distanciaBC));
32     //Calcule primero la segunda forma ya que me parecio sencillo de primera mano, asi que ya la use para comprobar si de las dos maneras coincidian
33     double areaSegundaForma = 0.25 * Math.sqrt ( Math.pow( Math.pow (a: distanciaAB,b: 2) + Math.pow(a: distanciaAC,b: 2) + Math.pow(a: distanciaBC,b: 2), b: 2)
34         - 2 * (Math.pow(a: distanciaAB, b: 4) + Math.pow(a: distanciaAC, b: 4) + Math.pow(a: distanciaBC, b: 4)));
35
36     String cadena = ("Punto A : (" + ParamX1 + " , " + ParamY1 + ") \n" +
37         "Punto B : (" + ParamX2 + " , " + ParamY2 + ") \n" +
38         "Punto C : (" + ParamX3 + " , " + ParamY3 + ") \n \n" +
39         "La distancia entre A Y B : " + distanciaAB + " \n" +
40         "La distancia entre A Y C : " + distanciaAC + " \n" +
41         "La distancia entre B Y C : " + distanciaBC + " \n \n" +
42         "El area del triangulo formado por los puntos es : \n" +
43         "Area = (" + semiperimetro + " (" + semiperimetro + " - " + distanciaAB + ") (" + semiperimetro + " - " + distanciaAC +
44         ") (" + semiperimetro + " - " + distanciaBC + ") )" + SimboloElevado + "1/2 =" + areaPrimeraForma );
45
46     return cadena;
47 }

```

(Creación de métodos)

-Los nombres de los métodos tienen significado y puede atisbarse la funcionalidad de cada uno, a su vez pueden ser pronunciados perfectamente, el único inconveniente es que los dos métodos sobrecargados no tienen un nombre que sea un verbo por lo cual cambiare el nombre de ellos a “CalcularDistancia” y a “CalcularArea”.

-Los métodos getter están perfectos, (funcionalidad pequeña y simple, hace una sola cosa, no hace abuso de ningún switch, tienen un único argumento, no tiene flag arguments, no generan efectos colaterales, no están repetidos en ningún lado).

El tercer método que se llamara “CalcularDistancia” pero ahora se llama distancia2D y coge un punto de parámetro, cumple también los requisitos anteriores por lo cual esta perfecto, pero podría simplificarse aun mas prescindiendo de la variable “distancia_calculada” y haciendo que el return sea directamente el calculo.

El cuarto método que se llamara “CalcularArea” pero ahora se llama distancia2D, es el que trae muchos problemas ya que no cumple una sola funcionalidad y tampoco es simple.

Por lo cual haré un nuevo método privado que sea invocado exclusivamente cuando se invoque este cuarto, ese método se encargara de devolverme un string que calcule y devuelva la distancia de los 3 puntos para poder quitarme las tres primeras lineas del método y del String cadena.

Ha su vez sacare la variable char SimboloElevado ya que puede prescindirse de ella por una cosa que diré mas tarde.

En el mismo método se puede ver un comentario que habla sobre la repetición de la obtención de un dato pero con otra formula, suprimiré la mas larga y me quedare con la mas corta ya que esta segunda no era ni devuelta a pantalla.

Haré también que el String cadena no exista y haré el return directamente con el texto (el cual voy a reducir su contenido que tiene cosas innecesarias como la formula) que quiero obtener con este método.

A medida que fui haciendo los cambios me di cuenta de que necesitaba un método mas que se encargaba de medir las distancias entre dos puntos, ya que con el método "CalcularDistancia" no podía hacerlo debido a sus parámetros.

-Aun que son unos métodos un poco mas complejos de lo que define el bloque dos de clean code, han quedado bastante ordenados y entendibles por lo cual considero que se puede prescindir completamente de comentarios en el código.

El Resultado

```
1 package daml_delafuentecedillo_jonathan_ej0501b10punto3d;
2
3 public class Punto {
4
5     public float coordenadaX;
6     public float coordenadaY;
7     private static int contador = 0;
8     private static String punto = "Punto";
9     final private int id;
10
11     public static int getContador() {
12         return contador;
13     }
14
15     public static String getPunto() {
16         return punto;
17     }
```

(Creación de la clase, declaración de variables y de dos métodos getter)

```

20     return Math.sqrt(Math.pow(Param.coordenadaX-coordenadaX, b: 2)+Math.pow(Param.coordenadaY-coordenadaY, b: 2));
21 }
22
23 public static String CalcularArea (double ParamX1, double ParamY1, double ParamX2, double ParamY2, double ParamX3, double ParamY3){
24     double semiperimetro = ((CalcularDistancia(ParamX1, ParamY1, ParamX2, ParamY2)
25     + (CalcularDistancia(ParamX1, ParamY1, ParamX2:ParamX3, ParamY2:ParamY3))
26     + (CalcularDistancia(ParamX1:ParamX2, ParamY1:ParamY2, ParamX2:ParamX3, ParamY2:ParamY3)))/2);
27
28     double areaPrimeraForma = Math.sqrt(semiperimetro * (semiperimetro-CalcularDistancia(ParamX1, ParamY1, ParamX2, ParamY2))
29     * (semiperimetro-CalcularDistancia(ParamX1, ParamY1, ParamX2:ParamX3, ParamY2:ParamY3))
30     * (semiperimetro-CalcularDistancia(ParamX1:ParamX2, ParamY1:ParamY2, ParamX2:ParamX3, ParamY2:ParamY3)));
31
32     return (DevolverTresDistancias(ParamX1, ParamY1, ParamX2, ParamY2, ParamX3, ParamY3) +
33     "El area del triangulo formado por los puntos es : \n" +
34     "Área = " + areaPrimeraForma );
35 }
36
37 private static double CalcularDistancia (double ParamX1, double ParamY1, double ParamX2, double ParamY2){
38     return Math.sqrt(Math.pow(ParamX1-ParamX2, b: 2)+Math.pow(ParamY1-ParamY2, b: 2));
39 }
40
41 private static String DevolverTresDistancias(double ParamX1, double ParamY1, double ParamX2, double ParamY2, double ParamX3, double ParamY3){
42     return "Punto A : (" + ParamX1 + " , " + ParamY1 + ")\n" +
43     "Punto B : (" + ParamX2 + " , " + ParamY2 + ")\n" +
44     "Punto C : (" + ParamX3 + " , " + ParamY3 + ")\n\n" +
45     "La distancia entre A Y B : " + CalcularDistancia(ParamX1, ParamY1, ParamX2, ParamY2) + "\n" +
46     "La distancia entre A Y C : " + CalcularDistancia(ParamX1, ParamY1, ParamX2:ParamX3, ParamY2:ParamY3) + "\n" +
47     "La distancia entre B Y C : " + CalcularDistancia(ParamX1:ParamX2, ParamY1:ParamY2, ParamX2:ParamX3, ParamY2:ParamY3) + "\n\n";
48 }
49

```

(Creación de métodos)

Aquí se puede observar las dos nuevas variables que simplifican el código de “CalcularArea”, repartíéndose las tareas entre las tres ahora tenemos métodos mas sencillos que el método anterior. Los nombres de las dos variables que estaban ya creadas están cambiadas

```

50 public Punto( float X , float Y){
51     this.coordenadaX = X;
52     this.coordenadaY = Y;
53     contador++;
54     id = contador;
55 }
56
57 public Punto() {
58     coordenadaX = 0f;
59     coordenadaY = 0f;
60     contador++;
61     id = contador;
62 }
63
64 public static void setPunto(String aPunto) {
65     punto = aPunto;
66 }
67
68 public String toString(){
69     String cadena = punto + " " + id + "(" + coordenadaX + " , " + coordenadaY + ")";
70     return cadena;
71 }
72
73 public void setX (float X){ this.coordenadaX = X; }
74 public float getX () { return coordenadaX; }
75 public void setY (float Y){ this.coordenadaY = Y; }
76 public float getY () { return coordenadaY; }
77
78 public int getId() {
79     return id;
80 }

```

(Creación de métodos usuales en una clase)

Aquí se puede ver el resto del código que pertenece a una clase normalmente es decir métodos getter, setter, constructores y toString. (Esto es para entender mejor la clase main)

```

1 package dam1_delafuentecedillo_jonathan_ej0501b10punto3d;
2
3 public class MetodoPrincipal {
4
5     public static void main(String[] args) {
6         Punto A = new Punto (x: 0.41f,-3.77f);
7
8         System.out.println("A " + A);
9
10        System.out.println(x: " \nSalida directa toString() :");
11        System.out.println(x: A.toString());
12
13        System.out.println("Hay un total de " + Punto.getContador() + " " + Punto.getPunto() + "s");
14
15        System.out.println(x: Punto.CalcularArea(ParamX1:3.24,ParamY1:4.81, ParamX2:0.41, -3.77, ParamX3:6.412, -1.423));
16
17    }
18
19 }

```

(Clase principal con el método main)

```

Output - DAM1_DeLaFuenteCedilloJonathan_CleanCode (run) X
run:
A Punto 1(0.41 , -3.77)

Salida directa toString() :
Punto 1(0.41 , -3.77)
Hay un total de 1 Puntos
Punto A : (3.24 , 4.81)
Punto B : (0.41 , -3.77)
Punto C : (6.412 , -1.423)

La distancia entre A Y B : 9.034672102517058
La distancia entre A Y C : 6.993702381428594
La distancia entre B Y C : 6.444564609032948

El area del triangulo formado por los puntos es :
Area = 22.427575000000008
BUILD SUCCESSFUL (total time: 0 seconds)

```

(Ejecución del programa)