

CLEAN CODE

Carolina Dranicer Dranicer

Partimos con el siguiente código, no es incorrecto pero su lectura no es sencilla.

```
public class CleanCode_CarolinaDranicerDranicer {

    public static void main(String[] args) {
        class Punto {
            float crdX;
            float crdY;
        }

        Punto ptP = new Punto();
        Punto ptQ = new Punto();
        Punto ptR = new Punto();
        ptP.crdX = 1.1f;
        ptP.crdY = 2.2f;
        ptQ.crdX = 4.7f;
        ptQ.crdY = 7.4f;
        ptR.crdX = 0.5f;
        ptR.crdY = 3.5f;
        System.out.printf(format:"Las coordenadas de los pts son:\n P(%.1f , %.1f)\n Q(%.1f , %.1f)\n R(%.1f , %.1f)\n"
            , args: ptP.crdX , args: ptP.crdY
            , args: ptQ.crdX , args: ptQ.crdY
            , args: ptR.crdX , args: ptR.crdY);

        System.out.printf(format:"La distancia del pt P a Q : %.2f \n" ,
            args: Math.sqrt((Math.pow(a: ptQ.crdX,b: 2)-Math.pow(a: ptP.crdX,b: 2)) + (Math.pow(a: ptQ.crdY,b: 2)-Math.pow(a: ptP.crdY,b: 2))));
        System.out.printf(format:"La distancia del pt P a R : %.2f \n" ,
            args: Math.sqrt((Math.pow(a: ptR.crdX,b: 2)-Math.pow(a: ptP.crdX,b: 2)) + (Math.pow(a: ptR.crdY,b: 2)-Math.pow(a: ptP.crdY,b: 2))));
        System.out.printf(format:"La distancia del pt R a Q : %.2f \n" ,
            args: Math.sqrt((Math.pow(a: ptQ.crdX,b: 2)-Math.pow(a: ptR.crdX,b: 2)) + (Math.pow(a: ptQ.crdY,b: 2)-Math.pow(a: ptR.crdY,b: 2))));
    }
}
```

NOMBRES

Vamos a cambiar los nombres de las variables siguiendo el principio de *Camel case* para que sean sencillas de pronunciar, tengan un significado propio y sean fáciles de buscar y encontrar.

```
public static void main(String[] args) {
    class Punto {
        float coordenadaX;
        float coordenadaY;
    }

    Punto P = new Punto();
    Punto Q = new Punto();
    Punto R = new Punto();
    P.coordenadaX = 1.1f;
    P.coordenadaY = 2.2f;
    Q.coordenadaX = 4.7f;
    Q.coordenadaY = 7.4f;
    R.coordenadaX = 0.5f;
    R.coordenadaY = 3.5f;
    System.out.printf(format:"Las coordenadas de los puntos son:\n P(%.1f , %.1f)\n Q(%.1f , %.1f)\n R(%.1f , %.1f)\n"
        , args: P.coordenadaX , args: P.coordenadaY
        , args: Q.coordenadaX , args: Q.coordenadaY
        , args: R.coordenadaX , args: R.coordenadaY);

    System.out.printf(format:"La distancia del punto P a Q : %.2f \n" ,
        args: Math.sqrt((Math.pow(a: Q.coordenadaX,b: 2)-Math.pow(a: P.coordenadaX,b: 2)) + (Math.pow(a: Q.coordenadaY,b: 2)-Math.pow(a: P.coordenadaY,b: 2))));
    System.out.printf(format:"La distancia del ppunto P a R : %.2f \n" ,
        args: Math.sqrt((Math.pow(a: R.coordenadaX,b: 2)-Math.pow(a: P.coordenadaX,b: 2)) + (Math.pow(a: R.coordenadaY,b: 2)-Math.pow(a: P.coordenadaY,b: 2))));
    System.out.printf(format:"La distancia del punto R a Q : %.2f \n" ,
        args: Math.sqrt((Math.pow(a: Q.coordenadaX,b: 2)-Math.pow(a: R.coordenadaX,b: 2)) + (Math.pow(a: Q.coordenadaY,b: 2)-Math.pow(a: R.coordenadaY,b: 2))));
}
}
```

FUNCIONES

Las operaciones se expresan en el print, por lo que vamos a crear una única función en la clase donde se realicen las operaciones indicadas así como el String para que el código no sea repetitivo y más pequeño el print, esta tendrá un nombre significativo.

```
public class CleanCode_CarolinaDranicerDranicer {  
  
    public static void main(String[] args) {  
        class Punto {  
            float coordenadaX;  
            float coordenadaY;  
  
            public Punto (float CoordenadaX, float CoordenadaY){  
                this.coordenadaX = CoordenadaX;  
                this.coordenadaY = CoordenadaY;  
            }  
            public String toString(){  
                String retorno = "(" + coordenadaX + " , " + coordenadaY + ")";  
                return retorno;  
            }  
            public double distancia(Punto Param){  
                double distancia_calculada = Math.sqrt (Math.pow(Param.coordenadaX-coordenadaX, b: 2)+Math.pow(Param.coordenadaY-coordenadaY, b: 2));  
                return distancia_calculada;  
            }  
        }  
  
        Punto P = new Punto(CoordenadaX: 1.1f , CoordenadaY: 2.2f);  
        Punto Q = new Punto(CoordenadaX: 4.7f , CoordenadaY: 7.4f);  
        Punto R = new Punto(CoordenadaX: 0.5f , CoordenadaY: 3.5f);  
        System.out.println("Las coordenadas de los puntos son:\nP:" + P.toString() + " Q:" + Q.toString() + " R:" + R.toString());  
        System.out.printf(format:"La distancia del punto P a Q : %.2f \n" , args: P.distancia(Param: Q));  
        System.out.printf(format:"La distancia del punto P a R : %.2f \n" , args: P.distancia(Param: R));  
        System.out.printf(format:"La distancia del punto R a Q : %.2f \n" , args: R.distancia(Param: Q));  
    }  
}
```

COMENTARIOS

Para finalizar vamos a añadir algún comentario para explicar que hace el código, aunque ya este código sea autoexplicativo.

```
public class CleanCode CarolinaDranicerDranicer {  
  
    public static void main(String[] args) {  
        class Punto {  
            float coordenadaX;  
            float coordenadaY;  
  
            public Punto (float CoordenadaX, float CoordenadaY){  
                this.coordenadaX = CoordenadaX;  
                this.coordenadaY = CoordenadaY;  
            }  
            public String toString(){  
                String retorno = "(" + coordenadaX + " , " + coordenadaY + ")";  
                return retorno;  
            }  
            public double distancia(Punto Param){  
                double distancia_calculada = Math.sqrt (Math.pow(Param.coordenadaX-coordenadaX, b: 2)+Math.pow(Param.coordenadaY-coordenadaY, b: 2));  
                return distancia_calculada;  
                //Formula matematica que calcula distancia entre puntos 2D  
            }  
        }  
  
        //Declaracion de puntos  
        Punto P = new Punto(CoordenadaX: 1.1f , CoordenadaY: 2.2f);  
        Punto Q = new Punto(CoordenadaX: 4.7f , CoordenadaY: 7.4f);  
        Punto R = new Punto(CoordenadaX: 0.5f , CoordenadaY: 3.5f);  
        //Muestra por pantalla las coordenadas de los distintos puntos  
        System.out.println("Las coordenadas de los puntos son:\nP:" + P.toString() + " Q:" + Q.toString() + " R:" + R.toString());  
        //Muestra por pantalla de las distancias  
        System.out.printf(format:"La distancia del punto P a Q : %.2f \n" , args: P.distancia(Param: Q));  
        System.out.printf(format:"La distancia del punto P a R : %.2f \n" , args: P.distancia(Param: R));  
        System.out.printf(format:"La distancia del punto R a Q : %.2f \n" , args: R.distancia(Param: Q));  
    }  
}
```