

Práctica 6 opcional

Objetivo: Practicar recorridos y búsquedas en secuencias

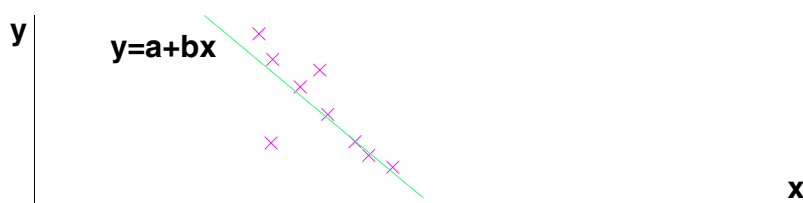
Descripción: Escribir una clase llamada **ListaPuntos** para realizar cálculos con una lista de parejas de valores reales

La clase dispondrá de los siguientes atributos:

- **lista:** un objeto de la clase **CajaTexto** del paquete **fundamentos**, para almacenar la lista de puntos
 - en la ventana de este objeto, el usuario tecleará líneas con dos números reales en cada una, representando las coordenadas x e y de un punto en el plano
- **regresion:** un objeto de la clase **RegresionLineal**
 - consultar en **Bluej** la documentación de la clase **RegresionLineal**

Práctica 6 opcional (cont.)

El cálculo de regresiones lineales permite encontrar la recta que más se aproxima a la función $y(x)$ para los puntos medidos



La recta se determina en función de:

- coeficientes a y b , de la ecuación de la recta
- coeficiente de correlación, r , que mide la precisión de la solución
 - la solución es mejor cuanto más cerca de la unidad esté r

Práctica 6 opcional

La clase debe tener los siguientes métodos

- **calculaRegresion**: lee de la caja de texto las parejas de valores x e y y los inserta en el objeto **regresion** (con su método **inserta**)
 - usar el esquema de recorrido en secuencias para insertar puntos
- **distancias**: llama a **calculaRegresion** y luego retorna un objeto de la clase **CajaTexto** en el que se escriben los valores de las distancias de cada punto del objeto **lista** a la recta de regresión
 - la distancia de un punto (x,y) a la recta se define como $y-a+bx$, siendo a y b los coeficientes de la recta de regresión, obtenidos con los métodos **coefA** y **coefB**
 - usar el esquema de recorrido en secuencias para recorrer todos los puntos y calcular sus distancias

Práctica 6 opcional

Métodos de la clase (continuación)

- **dibujaRegresion**: llama a **calculaRegresion** y después muestra el dibujo de la regresión lineal (con **dibuja**);
- **distanciaExcesiva**: llama a **calculaRegresion** y luego retorna un booleano que indica si la distancia de alguno de los puntos de **lista** a la recta de regresión excede en valor absoluto de un valor real que se pasa como parámetro
 - retorna **true** si al menos una de las distancias es excesiva (en valor absoluto) y **false** si ninguna es excesiva
 - seguir el esquema de búsqueda en secuencias

Realización: Codificar la clase en Java y probar todos los métodos

Entregar: El código de la clase **ListaPuntos** en un archivo **.jar**