

Problema 5: Practicar recorridos y búsquedas en arrays y ArrayList

Datos personales	
Apellidos:	Cédric
Nombre:	Prieels

1 Convertir pseudocódigo a código

Objetivos:

Practicar el paso de pseudocódigo a código para calcular una regresión logarítmica a partir de dos arrays de datos reales.

Descripción:

Escribir en Java el siguiente algoritmo descrito en pseudocódigo, que calcula los datos de una regresión logarítmica a partir de dos arrays de puntos, $x[]$ e $y[]$, del mismo tamaño. El método calcula los coeficientes a y b de la siguiente curva con la que se intenta ajustar el conjunto de puntos x - y :

$$y = a \ln(x) + b$$

Donde $\ln(x)$ es el logaritmo neperiano de x . Los datos calculados se almacenarán en los atributos reales a y b , pertenecientes a la misma clase que el método. Si x e y tienen distinto tamaño se indicará el error poniendo a a y b a valor NaN (*Not a Number*).

```
método regresionLogaritmica (array de reales x, array de reales y)
  si x e y tienen distinto tamaño entonces
    a=NaN
    b=NaN
  si no
    real sumX=0, sumY=0, sumLnX=0, sumLn2X=0, sumLnXY=0, sumY2=0
    entero n= longitud de x
    para i desde 0 hasta n -1
      sumX= sumX+x[i]
      sumY= sumY+y[i]
      sumLnX= sumLnX+ln(x[i])
      sumLn2X= sumLn2X+ln2(x[i])
      sumLnXY= sumLnXY+ln(x[i])*y[i]
      sumY2= sumY2+(y[i])2
    fin para
    a=(sumLnXY – sumY*sumLnX/n)/(sumLn2X – sumLnX*sumLnX/n)
    b=sumY/n – a*sumLnX/n
  fin si
fin método
```

La regresión logarítmica es una técnica que permite obtener los coeficientes de la curva logarítmica $y = a \ln(x) + b$ que más se aproximan a la función $y=f(x)$, definiéndose esta función como un conjunto de parejas (x,y) . Fuente: https://es.wikipedia.org/wiki/Regresi%C3%B3n_no_lineal

Respuesta que se pide

```
public static void main regresionLogaritmica(double[] x, double[] y) {
    double a, b = 0.0; // Esto no está escrito en el pseudocodigo pero siempre es mejor escribirlo.
    if (x.length != y.length) {
        a = Double.NaN;
        b = Double.NaN;
    }
    else {
        double sumX = 0.0, sumY = 0.0, sumLnX = 0.0;
        double sumLn2X = 0.0, sumLnXY=0.0, sumY2 = 0.0;

        int n = x.length;
        for (int i = 0; i <= (n-1); i++) {
            sumX += x[i];
            sumY += y[i];
            sumLnX += Math.log(x[i]);
            sumLn2X += Math.pow(Math.log(x[i]),2);
            sumLnXY += Math.log(x[i]) * y[i];
            sumY2 += Math.pow(y[i],2);
        }
        a=(sumLnXY – sumY*sumLnX/n)/(sumLn2X – sumLnX*sumLnX/n);

        b=(sumY/n – a*sumLnX/n);
    }
    // No sé si hay que añadir además un (this.a = a), para hacer algo con los atributos. No esta
    // escrito en el pseudocodigo asi que prefiero no poner nada.
}
```

2 **Búsqueda en un array**

Objetivos

Practicar la búsqueda en un array y describirla mediante pseudocódigo y código.

Descripción

Se dispone de una clase llamada Asignatura y que implementa el siguiente diagrama de clases

Asignatura
-String nombre -String codigo -int numAlumnos
+Asignatura (String nombre, String codigo, int numAlumnos) +String getNombre() +String getCodigo() +int getNumAlumnos()

Como se puede ver es una clase trivial, con un constructor al que se le pasan los valores iniciales de los atributos, y tres métodos observadores, uno por cada atributo.

Se desea escribir un método estático al que se le pasan como parámetros un array de objetos de la clase Asignatura y un código. El método debe retornar el nombre de la asignatura del array cuyo código coincida con el indicado en el parámetro, o null si no se encuentra.

Respuesta que se pide

método busqueda devuelve un String

para i desde 0 hasta la longitud del array arrayAsignaturas (pasado como parametro) hacer
si elemento i es igual al codigo (otro parametro dado)

retornar el nombre de la asignatura

fin si

fin para

retornar valor "null" si hace falta

fin método

```
public static String main busqueda(Asignatura[] arrayAsignaturas, String codigo) {  
    for (int i=0; i < arrayAsignaturas.length; i++) {  
        if (arrayAsignaturas[i].getCodigo().equals(codigo)) {  
            return arrayAsignaturas[i].getNombre();  
        }  
    }  
}
```

```
        return "null";  
    }  
}
```

3 *Búsqueda en un ArrayList*

Objetivos

Hacer la misma búsqueda anterior en un ArrayList.

Descripción

Se desea escribir un método estático al que se le pasan como parámetros un ArrayList de objetos de la clase Asignatura de la cuestión anterior y un código. El método debe retornar el nombre de la asignatura de la lista cuyo código coincida con el indicado en el parámetro, o null si no se encuentra.

Respuesta que se pide

```
import java.util.ArrayList;  
  
public static String busqueda(ArrayList<Asignatura> listaAsignaturas, String codigo) {  
    String resultado = "null";  
    for (int i = 0; i < listaAsignaturas.size(); i++) {  
        if (listaAsignaturas.get(i).getCodigo().equals(codigo)) {  
            resultado = listaAsignaturas.get(i).getNombre();  
        }  
    }  
    return resultado;  
}
```

4 *Desviación estándar*

Objetivos

Calcular la desviación estándar de una serie de medidas

Descripción

Se desea escribir un método estático al que se le pasa como parámetro un array de n números reales y que calcula y retorna la desviación estándar obtenida con la expresión:

$$\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$$

donde:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

Programación, Curso 2015-2016

Nota: Observar que en las fórmulas los valores se numeran de 1 a n , mientras que en el array van de 0 a $n-1$

Respuesta que se pide

```
public static double calculoDesviacion(double[] numerosDados) {  
    int longitud = numerosDados.length;  
    double xMedia = 0.0, desviacion = 0.0;  
    for (int i=0; i<longitud; i++) {  
        xMedia += numerosDados[i]/longitud;  
    }  
    for (int i=0; i<longitud; i++) {  
        desviacion += Math.pow((numerosDados[i]-xMedia),2)/(longitud-1);  
    }  
    return desviacion;  
}
```