

# Programación en Lenguaje Java

## Práctica 10.1. Ajuste polinómico de un conjunto de puntos



**Michael González Harbour**  
**Mario Aldea Rivas**

Departamento de Matemáticas,  
Estadística y Computación

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

# Práctica 10-1: Ajuste polinómico de un conjunto de puntos

---

*Objetivos:* Practicar la entrada/salida de ficheros de texto, así como practicar con arrays, y con la manipulación de textos

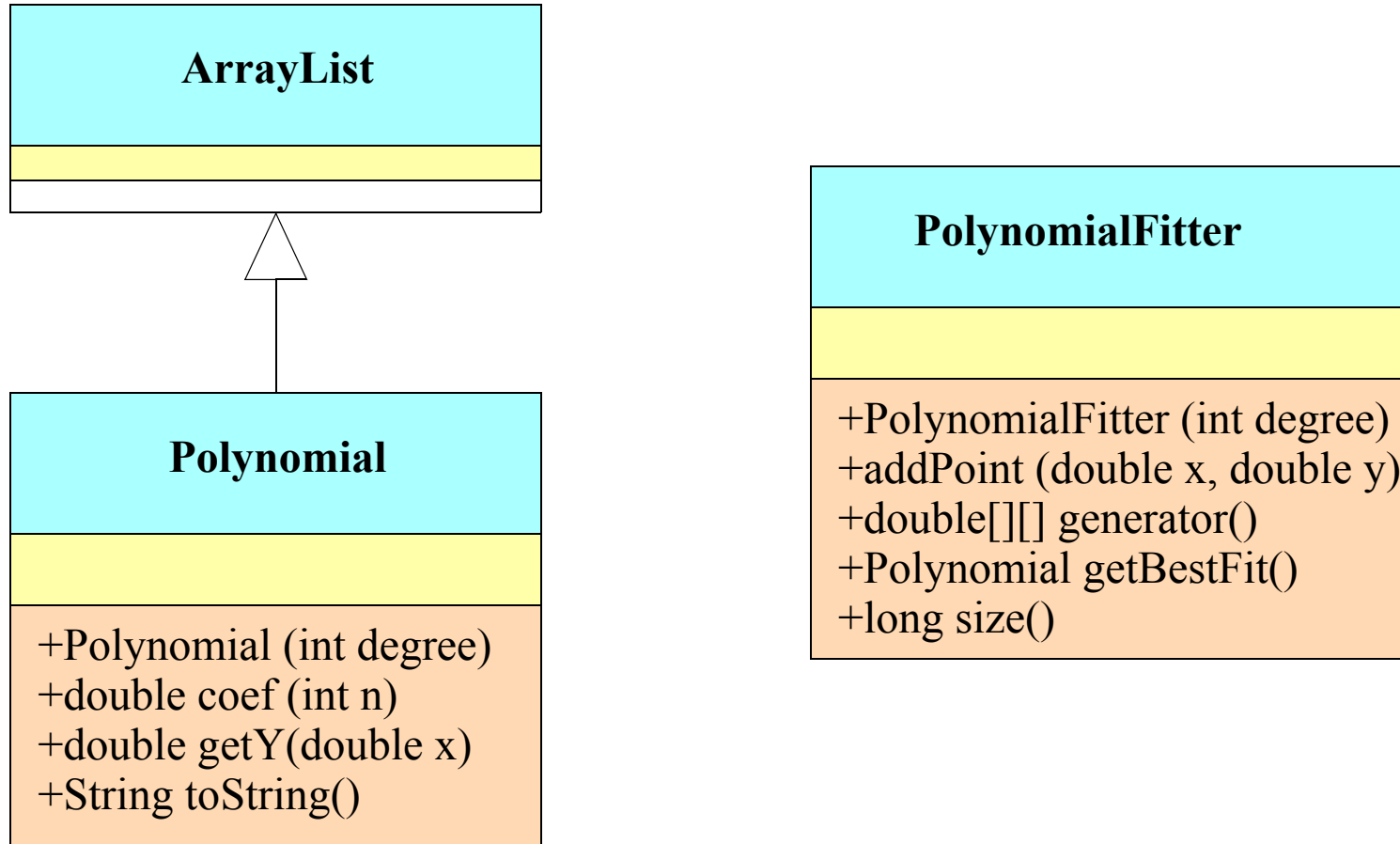
Se dispone de un par de clases ya hechas que nos permiten ajustar un conjunto de puntos  $(x,y)$  por medio de un polinomio de orden  $n$  de la forma

$$y = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

*Descripción:* Escribir un programa para ajustar unos puntos mediante un polinomio de orden  $n$ , leyendo los puntos  $(x,y)$  de un fichero de texto

# Clases disponibles

---



# Clases disponibles (cont.)

---

Para poder realizar el ajuste se dispone de las clases `Polynomial` y `PolynomialFitter`, cuyos diagramas de clase (en su parte accesible) se han mostrado arriba

Para conocer su funcionamiento se puede consultar la documentación de las clases

Si se desea obtener más información sobre el algoritmo de regresión polinómica, se puede consultar la dirección:

<http://www.arachnoid.com/polysolve/>

# Descripción

---

Se escribirá un programa que haga:

1. Leer por teclado el orden del polinomio a ajustar
2. Crear un objeto de la clase `PolynomialFitter`
3. Leer los datos del fichero `EuriborA12Meses.csv` mediante el método `leeDatos()` que hay que escribir
  - este método tendrá como parámetro el nombre del fichero
  - este método retornará los datos en una matriz de dos columnas, la primera para los valores  $x$ , la segunda para los valores  $y$
4. Representar gráficamente los datos junto con el ajuste
5. Escribir en pantalla los coeficientes del polinomio ajustado, usando notación de punto fijo con 6 decimales

Utilizar el pseudocódigo que se describe a continuación

# Pseudocódigo del programa principal

---

```
// leer el orden del polinomio
entero orden
leer de teclado orden
// crear variables xmin y xmax y el ajustador
real xmin = máximo valor real
real xmax = mínimo valor real
PolynomialFitter p= nuevo PolynomialFitter(orden)
// Leer los datos del fichero con el método leeDatos
real[][] datos=leeDatos("EuriborA12Meses.csv");
// crear la gráfica recorrer todos los puntos
Grafica g = nueva Grafica
para i desde 0 hasta longitud de datos -1 hacer
    real x=datos[i][0]
    real y=datos[i][1]
    // insertar x e y en la gráfica y en el ajustador
    g.inserta(x, y)
    p.addPoint(x,y)
    // calcular el mínimo y máximo de las x
    si x<xmin entonces xmin=x; fin si
    si x>xmax entonces xmax=x; fin si
fin para
```

# Pseudocódigo (cont.)

---

```
// calcular el ajuste polinómico
Polynomial resul=p.getBestFit()
// Mostrar en pantalla los coeficientes
para i desde orden hasta 0 paso -1 hacer
    muestra en pantalla resul.coef(i)
fin para
// hacer la gráfica del polinomio resultante
g.otraGrafica()
para x desde xmin hasta xmax paso (xmax-xmin)/200 hacer
    g.inserta(x,resul.getY(x))
fin para
g.pinta()
```

# Formato del fichero

---

El fichero de datos se llama `EuriborA12Meses.csv` y debe incluirse en el proyecto de BlueJ

Los datos están separados con punto y coma, con la siguiente estructura:

```
Fecha;valorFecha;Euribor  
02/01/09;9.0027397;3.025  
05/01/09;9.0109589;2.995
```

...

La segunda columna de datos contiene la fecha convertida en un valor numérico (en años desde 2000)

Usaremos la segunda columna para obtener los valores  $x$  y la tercera para los  $y$



# Método leeDatos

---

El fichero se puede leer línea a línea, cada línea en un String. Será necesario leer el fichero dos veces:

- una para contar el número de líneas y así poder crear el array
- la segunda para leer los datos y meterlos en el array

Cada línea leída se puede separar en sus tres datos con el método `split()` de la clase `String`

```
String[] split  
    (String separador)
```

```
Retorna un array de Strings con los  
fragmentos delimitados por el separador  
que recibe como parámetro
```

El primer dato se ignora. Los otros dos se convierten a número con el método `parseDouble()` de la clase `Double`

```
static double parseDouble  
    (String str)
```

```
Retorna el resultante de la conversión de  
str a número real. Si la conversión no es  
posible lanza NumberFormatException.
```

# Informe

---

## Entregar:

- El código Java de la clase que contiene el programa y el método `LeeDatos()`
- Captura de pantalla del ajuste realizado, indicando el grado del polinomio utilizado para el ajuste
- Los valores de los coeficientes obtenidos

# Parte avanzada

---

Modificar el programa y el método `leeDatos()` para que trate adecuadamente los siguientes errores que pueden producirse:

- Error de tipo `NumberFormatException`
  - tanto en la lectura por pantalla
  - como en la lectura desde el fichero de datos
- El número de datos de la línea no es tres

*Entregar.* El código Java del nuevo programa