

Práctica 5 – Programando en Java (2)

1- Objetivos

El objetivo de esta práctica es avanzar con la programación en Java, adquiriendo las habilidades necesarias para hacer programas que procesen ficheros, leyendo la información que contienen y realizando cálculos con ella. Para conseguirlo se realizará un programa que analice una lista de números en un fichero obteniendo el histograma y estadísticas similares.

2- Leyendo un fichero y escribiendo a la salida estándar [tiempo estimado 50min]

En esta práctica se pretende realizar un programa que lea una lista de números de un fichero y realice operaciones con ellos. Como primer paso asegúrese de que puede leer números enteros de un fichero realizando un programa con las siguientes especificaciones:

Uso:

```
java LeeEnteros <nombredelfichero> <numeroreferencia>
```

Lee un fichero con el nombre indicado con líneas que contienen un número entero. Por cada número que lea debe imprimir una línea que diga
Leido numero [x] es [menor|mayor|igual] que [numeroreferencia]

Ejemplos:

```
$ cat fichero1  
1  
4  
1  
15  
21  
1  
2  
4  
$ java LeeEnteros fichero1 4  
Leido [1] es menor que [4]  
Leido [4] es igual que [4]  
Leido [1] es menor que [4]  
Leido [15] es mayor que [4]  
Leido [21] es mayor que [4]  
Leido [1] es menor que [4]  
Leido [2] es menor que [4]  
Leido [4] es igual que [4]  
$
```

Guarde este programa para enseñarlo en el checkpoint.

3- Calculando el histograma [tiempo estimado 80min]

En esta parte se pretende realizar un programa que lea un fichero como el anterior que contenga números enteros no negativos (es aceptable que acabe con un error si hay algún número negativo o algo que no sea un número en una línea). El programa proporcionará el histograma analizando cuántas veces aparece cada número en el fichero. El programa aceptará como argumento un número máximo a partir del cual no nos interesa saber las repeticiones individuales. El funcionamiento del programa debe cumplir las siguientes especificaciones.

Uso:

```
java Histograma <nombrfichero> <maximonumero>
```

Calcula el histograma de los números que aparecen en el fichero indicado por nombrefichero. El histograma es el número de repeticiones de cada número individual. Solo nos interesa saber las repeticiones de los números menores que maximonumero. Los valores que no están dentro del rango nos vale con saber el número global. El fichero de entrada puede tener potencialmente un número de líneas ilimitado.

La salida será una línea por cada número de interés indicando el número y la cantidad de apariciones en el fichero. El formato de la salida debe ser el que se ve en el ejemplo

Ejemplos:

```
$ cat fichero1
1
4
1
15
21
1
2
4
$ java Histograma fichero1 10
0 : 0 veces
1 : 3 veces
2 : 1 veces
3 : 0 veces
4 : 2 veces
5 : 0 veces
6 : 0 veces
7 : 0 veces
8 : 0 veces
9 : 0 veces
otros : 2 veces
total : 8
$
```

Para calcular el histograma no es aceptable leer todos los números del fichero y después hacer los cálculos, ya que el fichero podría tener millones de números. Se trata de leer cada línea, utilizar la información de esa línea y olvidarse de ese número. Es un requisito del programa que sea capaz de analizar un fichero de 1000000 de elementos. Si intenta leerlos todos fallará.

Se recomienda utilizar el número de referencia para decidir cuántos contadores necesitamos, crear los contadores necesarios y para cada número leído actualizar los contadores correspondientes. Una vez leídos todos los números se imprimirá el resultado.

CHECKPOINT 1: muestre al profesor que su programa funciona (puntuación 2%)

Muestre al profesor de prácticas casos de uso de su programa con ficheros de entrada diferentes y que calculen el resultado correcto. Si no ha superado el tiempo recomendado puede hacer la sección siguiente y mostrar también esos casos en la demostración.

Suba el código fuente de su programa en un fichero Histograma.java en la página web de la asignatura (puntuación 2%)

4- Probando y depurando...

Compruebe que su programa funciona correctamente con ficheros de ejemplo. Asegúrese de que el programa funcione con ficheros de entrada suficientemente grandes. Para ello se proporciona en la web de la asignatura el programa PoissonGenerator.java. Descárguelo y compílelo para poder usarlo.

Uso:

```
java PoissonGenerator <n> <lambda>
```

Escribe por la salida estándar el número de muestras indicado por el parámetro `n` de una variable aleatoria de Poisson con una media de llegadas indicada por `lambda`. Nótese que aunque `lambda` no sea entero siempre hay un número entero de llegadas. Por ejemplo, si genera un proceso con una media de 3.5 llegadas, será muy probable que lleguen 3 en algunos experimentos y 4 en otros. Tampoco quiere decir que siempre vayan a generarse la media de llegadas por experimento.

Ejemplos:

```
$ java PoissonGenerator 3 10  
8  
11  
14
```

Para guardar un 1000 muestras en un fichero puede hacer

```
$ java PoissonGenerator 1000 5 >fichero1
```

Utilice el programa PoissonGenerator para generar ejemplos para probar su programa. Compruebe con eso que su programa es capaz de calcular correctamente el histograma de ficheros con 1000, 10000 y 1000000 de muestras.

5- Modificando el programa

Finalmente demuestre que domina su programa como para hacer las modificaciones que se piden. Realice las modificaciones necesarias para que el programa cumpla las siguientes especificaciones.

Uso:

```
java Histograma2 <nombrefichero> <maximonumero>
```

Calcula el histograma de los números que aparecen en el fichero indicado por nombrefichero. Para cada valor se proporciona el número de apariciones y la fracción que representa respecto al número total de elementos. El fichero de entrada puede tener potencialmente un número de líneas ilimitado. La salida será una línea por cada número de interés indicando el número y la cantidad de apariciones en el fichero.

Al final del histograma el programa debe indicar el valor más frecuente. En el caso de que varios casos compartan el máximo número de repeticiones deberán enumerarse todos los valores separados por espacios. El formato de la salida debe ser el que se ve en el ejemplo

Ejemplos:

```
$ cat fichero1
1
4
1
15
8
6
2
4
$ java Histograma2 fichero1 10
0 : 0 0
1 : 3 0.250
2 : 1 0.125
3 : 0 0
4 : 2 0.250
5 : 0 0
6 : 1 0.125
7 : 0 0
8 : 1 0.125
9 : 0 0
otros : 1 0.125
total : 8
max : 1 4
$
```

CHECKPOINT 2: suba la modificación en la página de la asignatura (puntuación 2%)

Realice la modificación indicada y envíela en un fichero llamado Histograma2.java que debe compilar y funcionar correctamente.