

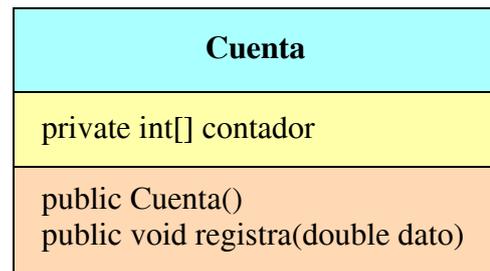
Examen Final de Fundamentos de Computadores y Lenguajes (Licenciado en Física)

Junio 2008

Primera parte (5 cuestiones, 50% nota del examen)

- 1) Escribir una clase que corresponde al diagrama de clases de la figura, en la que el constructor crea el array `contador` con cinco casillas, y el método `registra` incrementa la casilla `i` del array `contador`, siendo `i` correspondiente al rango en que se encuentra el valor de `dato`, según la siguiente tabla:

<code>i</code>	rango
0	$(-\infty, 0)$
1	$[0, 1)$
2	$[1, 10)$
3	$[10, 100)$
4	$[100, +\infty)$



- 2) Escribir un método al que se le pase como parámetro un texto, y que retorne un número real obtenido al convertir la última palabra del texto a número real. La última palabra está formada por los caracteres posteriores al último espacio en blanco, o por todo el texto si no hay ningún espacio en blanco. Si falla la conversión a número, retornar el número `Double.NaN`.
- 3) Escribir un método que utilice la ecuación de Van Der Waals para calcular y retornar el valor de la presión de un gas, P , dados el volumen molar, V_m , la temperatura, T , y la presión, temperatura y volumen críticos, P_c , T_c , y V_c . Las unidades de presión son atmósferas, el volumen litros, volumen molar litros/mol, y la temperatura grados Kelvin. El método debe seguir la cabecera:

```
public double presionVanDerWaals
(double Vm, double T, double Pc, double Tc, double Vc)
```

$$\left(P + \frac{a}{V_m^2}\right)(V_m - b) = RT$$

$$a = 3P_c V_c^2$$

$$b = \frac{V_c}{3}$$

$$R = \frac{8P_c V_c}{3T_c}$$

- 4) Queremos escribir un método que lee de un fichero binario números reales (del tipo `double`) y va almacenándolos en las casillas de un array de números reales que se le pasa como parámetro. Suponer que el fichero tiene suficientes números reales para rellenar todo el array. Si el fichero no existe se lanzará de modo automático la excepción `FileNotFoundException` (no hay que tratarla). Sí tratar otros errores que puedan producirse, escribiendo en pantalla un mensaje de error. El nombre del fichero se pasa como parámetro. El método debe obedecer a la siguiente cabecera:

```
public void lee(double[] datos, String nombreFichero)
    throws FileNotFoundException
```

- 5) Disponemos de un intérprete de órdenes en Linux en el que el directorio de trabajo se llama `/home/alumnos/pedro`. Queremos borrar de la carpeta `proyecto1` que está dentro del directorio de trabajo todos los ficheros que no tengan la extensión `.java`. Para ello podemos crear un directorio llamado `tmp` en `pedro` (suponer que `tmp` no existe), mover a él todos los ficheros de `proyecto1` acabados en `.java`, borrar el resto de ficheros de `proyecto1` y, finalmente, mover todos los ficheros de `tmp` a `proyecto1`. Indicar las instrucciones necesarias para ello.

Examen de Fundamentos de Computadores y Lenguajes (Licenciado en Física)

Junio 2008

Segunda parte (5 puntos, 50% nota del examen)

Se desea hacer una clase llamada `Medidas` para el tratamiento de las medidas obtenidas en un experimento, muestreando una variable a intervalos periódicos. Las medidas realizadas se hallan en un array que se pasa al constructor, junto al periodo de muestreo (tiempo, en segundos). El diagrama de clases de la figura muestra los atributos de la clase y las cabeceras de sus métodos, que debemos implementar según la siguiente descripción:

Medidas
<pre>private double[] dato private double periodo</pre>
<pre>public Medidas (double[] medida, double perMuestreo) public double media() throws NoHaySuficientesDatos public double frecuenciaMedia() throws NoHaySuficientesDatos public int flancoSiguiente(int i)</pre>

- **Constructor.** Debe crear el array `dato` del mismo tamaño que el parámetro `medida`, y luego copiar en `dato` todos los elementos de `medida`, uno por uno. Además debe copiar el parámetro `perMuestreo` en el atributo `periodo`.
- `media()`: calcula y retorna el valor medio de todos los datos guardados en el array `dato`. Si el array fuese de tamaño cero, lanzar `NoHaySuficientesDatos`.
- `frecuenciaMedia()`. Calcular y retornar la frecuencia media de las medidas, definida como el número de flancos positivos almacenados, dividido entre el tiempo total durante el que se tomaron las medidas. El tiempo total es el producto del periodo de muestreo por el número de medidas del array `dato`. Se considera que hay un flanco positivo cuando se encuentra en el array `dato` un valor negativo seguido en la siguiente casilla por un valor cero o positivo. Si se encuentran menos de 10 flancos positivos lanzar `NoHaySuficientesDatos`, pues se considera que el resultado no tiene suficiente precisión.
- `flancoSiguiente()`. Calcula y retorna la primera casilla del array `dato` que es posterior al parámetro `i` y en la que se encuentra un flanco positivo. Es decir, que esa casilla tiene un valor negativo, y la siguiente es cero o positiva. Si no hay ninguna casilla que cumpla esa condición retornar `-1`.

Nota: se valorarán los métodos a desarrollar de la siguiente forma:

- constructor: 2 puntos
- `media`: 1 punto
- `frecuenciaMedia`: 4 puntos
- `flancoSiguiente`: 3 puntos