

Examen de Programación (Grados en Física y Matemáticas)

Septiembre 2013

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

- 1) Se desea escribir un método al que se le pasa un array con números enteros. El método debe contar el número de unos, doses, treses y cuatros contenidos en el array (otros números se ignorarán). Luego retornará la media de estas cuatro cuentas. Por ejemplo, si hay 10 unos, 20 doses, 30 treses y 0 cuatros, se retornará $(10+20+30+0)/4$ pero calculado como número real, para no perder la parte fraccionaria.

La cabecera del método será:

```
public double mediaGrupos(int[] grupos)
```

- 2) Escribir el código Java de un método recursivo que muestra en pantalla los cuadrados de los primeros n números enteros positivos, separados por comas, siendo n el parámetro que se le pasa al método. El caso directo se da cuando $n=1$ y en ese caso basta mostrar en pantalla el número 1 seguido de una coma. En el caso recursivo se invoca al mismo método para $n-1$ y luego se muestra el valor de n^2 seguido de una coma.
- 3) Escribir el *pseudocódigo* de un método iterativo que calcula y retorna el desarrollo en serie de la función:

$$f(x) = e^{-x^2} = \sum_{i=0}^n (-1)^i \cdot \frac{x^{2i}}{i!}$$

El método recibe como parámetros los valores x y n . Para hacer más eficiente el cálculo, observar que los valores del factorial $i!$ y de la potencia x^{2i} se pueden obtener en la iteración número i para la iteración siguiente multiplicando el factorial anterior por $i+1$ y la potencia anterior por x^2 . Por su parte el signo se puede calcular con una variable que inicialmente vale 1 y a la que se cambia de signo a cada paso.

- 4) Escribir un método que lee todas las líneas de un fichero de texto cuyo nombre se pasa como parámetro y las escribe en pantalla, exceptuando las líneas vacías y las que comiencen por el carácter '#'.
5) En un computador con sistema operativo Linux se dispone de un directorio llamado medidas situado en el directorio del usuario. A su vez, este directorio tiene 4 subdirectorios, llamados tmp, val, cmd y resp.

Se desea escribir un *script* que se ejecute desde el directorio del usuario y que haga una copia selectiva de algunos ficheros, siguiendo el siguiente procedimiento:

- crear en el directorio del usuario un directorio llamado copia
- dentro de copia crear los directorios tiempos y valores
- copiar dentro de tiempos todos los archivos que estén en el directorio tmp
- mover todos los archivos acabados en ".dat" que estén en val al directorio valores
- copiar el directorio resp con todos sus contenidos dentro de copia

- cambiar de nombre al directorio resp que está dentro de medidas para llamarlo bck
- borrar todos los contenidos del directorio val, pero no el propio directorio.
- borrar el directorio cmd con todos sus contenidos

Dibujar además un esquema de la estructura del sistema de ficheros antes de ejecutar el *script*, y después.

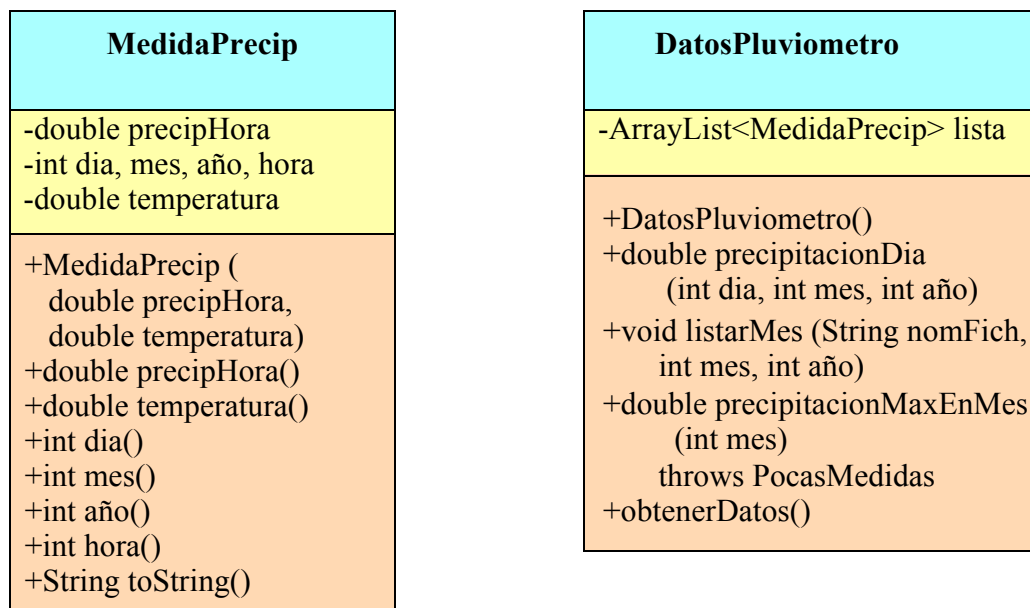
Examen de Programación (Grados en Física y Matemáticas)

Septiembre 2013

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

Se desea escribir parte del software de análisis de las medidas obtenidas por un pluviómetro. Se dispone de una clase llamada `MedidaPrecip` que contiene una medida de precipitación recogida en una hora (en litros/m²), así como la temperatura (en °C) y la hora, día, mes y año.

La figura muestra un diagrama de clases. Los métodos de `MedidaPrecip` son observadores de los distintos atributos, excepto `toString()` que retorna un texto que contiene todos los datos de la medida.



Lo que se pide es implementar en Java parte de la clase `DatosPluviometro` cuyo diagrama de clases se muestra arriba. La clase dispone de un único atributo que es una lista de objetos de la clase `MedidaPrecip`. Los métodos de esta clase hacen lo siguiente:

- `DatosPluviometro()`: Crea la lista vacía.
- `precipitacionDia()`: recorre la lista en busca de medidas de precipitación correspondientes al día, mes y año indicados por los parámetros. Suma las precipitaciones de todas estas medidas (que corresponden a diferentes horas) y retorna esa suma.
- `precipitacionMaxEnMes()`: Calcula y retorna la precipitación por hora máxima de todos los datos de la lista que correspondan al mes indicado en el parámetro (para todos los años). Si no hay ningún dato para ese mes lanza la excepción `PocasMedidas`.
- `listarMes()`: Escribe en el fichero de texto cuyo nombre se pasa como parámetro una lista de las precipitaciones de cada día correspondiente al mes y año indicados. Estas precipitaciones se obtienen con sucesivas llamadas al método `precipitacionDia()`. Por sencillez puede considerarse que todos los meses tienen 31 días. Los datos escritos en el fichero llevan una línea de título, una línea de cabecera, y 31 líneas con la precipitación

de cada día. A continuación se muestra un ejemplo de las primeras líneas de este fichero, que se usará como modelo para el formato. Respetar el tamaño de los datos y su alineación.

```
Precipitación del mes: 3, año: 2002
Día  Precipitación (l/m2)
1    0.0
2    0.2
3    23.3
...
```

- obtenerDatos(): Mete los datos en la lista. Este método no se pide.

Finalmente se debe escribir un programa principal en una clase aparte, que hace lo siguiente:

- crea un objeto de la clase DatosPluviometro
- invoca a su método obtenerDatos()
- muestra en pantalla la precipitación del 5 de septiembre de 2012
- muestra en pantalla la precipitación máxima en enero.
- escribe en el fichero "precip.txt" la precipitación de enero de 2012 llamando al método listarMes().

En este programa, si se lanza PocasMedidas se debe poner un mensaje de error en pantalla. El último paso (escribir en fichero) debe hacerse tanto si hay error como si no.

Valoración:

- constructor, atributo y encabezamiento de la clase: 0.5 puntos
- métodos: 1 punto cada uno
- programa principal: 1.5 puntos