

Examen de Programación (Grados en Física y Matemáticas)

Septiembre 2014

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

- 1) Escribir un método estático al que se le pasa como parámetro un string y que retorna la primera frase contenida en ese string, es decir, un string con todos los caracteres hasta el primer carácter '.' (excluido). Si no se encuentra ningún punto, retornar el string "****".
- 2) Escribir un método con la cabecera que se muestra abajo, que lea de un fichero de texto líneas y escriba en otro fichero de texto aquellas que *no* comiencen por el carácter '#'. Los nombres de los ficheros se pasan como parámetros. Utilizar la siguiente cabecera:

```
public static void filtra(String nombreFichEntrada, String nombreFichSalida)
```

- 3) Escribir un método que muestre en pantalla todos los caracteres del string que se le pasa como parámetro, uno por línea, comenzando por el último carácter del string y acabando por el primero.
- 4) Escribir el *pseudocódigo* de un método iterativo que calcula el siguiente desarrollo en serie de la función arcotangente:

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

Este desarrollo en serie solo funciona si x está entre -1.0 y 1.0.

El método recibe como parámetros los valores x y n . Si x no está entre -1.0 y 1.0 o n es menor que uno lanzar la excepción `DatosIncorrectos`. En caso contrario calcular y retornar el valor del sumatorio. Para hacer más eficiente el cálculo no utilizar la operación "elevar a". Observar que de la iteración i a la siguiente:

- el numerador cambia de signo alternando entre +1 y -1
- la potencia de x se obtiene multiplicando a la anterior por x^2

- 5) En un computador con sistema operativo Linux se ha montado una memoria USB en el directorio `/mnt/usbdisk1` y se desea escribir un script para manipular su información. El directorio de trabajo inicial es el directorio del usuario.

El *script* debe hacer lo siguiente:

- a. crear en el directorio del usuario un nuevo directorio llamado `datos` y dentro de él otros dos llamados `texto` y `resultados`
- b. copiar en `texto` todos los ficheros acabados en `.txt` que estén en la memoria USB.
- c. mover a `resultados` todos los ficheros acabados en `.res` que estén en la memoria USB
- d. borrar de la memoria USB todos los ficheros que acaben en `.dat`
- e. borrar de la memoria USB el directorio `backup` con todos sus contenidos

- f. cambiar de nombre al fichero thumbs.img de la memoria USB poniéndole el nombre imagenes.img

Hacer un dibujo con un esquema de la estructura de directorios resultante al final.

Examen de Programación (Grados en Física y Matemáticas)

Septiembre 2014

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

Se desea escribir parte del software de análisis de los datos obtenidos en experimentos de crecimiento de bacterias de diversas especies y en diversos medios. Los datos de cada experimento se guardan en un objeto de la clase Experimento cuyo diagrama se muestra abajo y que ya está implementada. Sus atributos son:

- especieBacteria: es un texto con la identificación de la bacteria
- medio: un texto que describe el medio en el que crece la bacteria
- poblacion6h: la población a las 6 horas en unidades por microlitro
- poblacion24h: la población a las 24 horas en unidades por microlitro

Como puede observarse, se dispone de un constructor al que se le pasan los datos del experimento. También hay métodos observadores, uno para cada atributo:

Experimento	ListaExperimentos
-String especieBacteria -String medio -double poblacion6h -double poblacion24h	-ArrayList<Experimento> lista -String persona
+Experimento (String especieBacteria, String medio, double poblacion6h, double poblacion24h) +String especieBacteria() +String medio() +double poblacion6h() +double poblacion24h()	+ListaExperimentos(String persona) +muestra() +Experimento mayorPoblacion(String tiempo) throws ErrorDeFormato +anade (Experimento e) throws Repetido

Lo que se pide es implementar en Java la clase ListaExperimentos cuyo diagrama de clases se muestra arriba. La clase dispone de los siguientes atributos:

- lista: guarda una lista de datos de experimentos
- persona: un String que contiene el nombre de la persona que hizo los experimentos.

Los métodos de la clase hacen lo siguiente:

- *constructor*: copia el parámetro en el atributo persona. Además, crea el atributo lista vacío.
- muestra(): muestra en pantalla todos los experimentos de la lista, con todos sus datos, uno por línea; se valorará poner los datos en columnas, así como poner una línea de encabezamiento. Si no hay experimentos, se pondrá un mensaje en pantalla indicándolo.

- `mayorPoblacion()`: retorna un objeto de la clase `Experimento` que es el que mayor población tiene de los almacenados en la lista a las 6h, si tiempo es igual a "6h", o a las 24h si tiempo vale "24h". Si tiempo no vale ninguno de esos valores se lanza `ErrorDeFormato`.
- `anade()`: Añade el experimento e a la lista siempre que no este repetido, es decir, que no exista en la lista un experimento con la misma especie de bacteria y el mismo medio. Si se encuentra que está repetido, se lanza `Repetido`.

Las excepciones `ErrorDeFormato` y `Repetido` están ya definidas en clases aparte y no es necesario hacerlas.

Finalmente, se pide hacer un programa principal de prueba en una clase aparte que haga lo siguiente:

- a. Crear un objeto de la clase `ListaExperimentos`
- b. Crear tres objetos de la clase `Experimento` y añadirlos a la lista
- c. Mostrar la lista en pantalla con el método `muestra()`
- d. Con el método `mayorPoblacion()` obtener las mayores poblaciones a las 6 horas y 24 horas. Mostrar en pantalla los datos de los experimentos obtenidos
- e. Añadir un experimento repetido para comprobar que salta la excepción
- f. Invocar el método `mayorPoblacion()` con un tiempo incorrecto para comprobar que salta la excepción

En este programa tratar las excepciones con mensajes de error. Si se produce alguna excepción, continuar con los siguientes pasos que se piden.

Valoración:

- encabezamiento de la clase, atributos y constructor: 0.5 puntos
- `muestra`: 1 punto
- `mayorPoblacion`: 1 punto
- `anade`: 1 punto
- programa principal: 1.5 puntos