

Examen de Fundamentos de Computadores y Lenguajes

Examen Parcial. Febrero 2006

Primera parte: Cuestiones (5 cuestiones, 5 puntos en total)

- 1) Crear una clase con dos atributos privados de la clase `String`. Uno de ellos permite almacenar el nombre de una persona, y el otro una frase que indica el grupo de edad. La clase tendrá un constructor al que se le pasan como parámetros un `String` con el nombre de la persona, y un entero con la edad. El constructor copia el nombre en el atributo correspondiente, y coloca la frase del grupo de edad al valor reflejado en esta tabla, según el intervalo al que pertenezca la edad

Grupo de edad	Intervalo de edades
"Niño"	0-14
"Joven"	15-22
"Adulto"	23 en adelante

- 2) Escribir un método que calcule y retorne el ángulo θ de un péndulo simple según la expresión

$$\theta = \theta_0 \cdot \text{sen}(\omega t + \varphi)$$

siendo θ_0 la amplitud del movimiento, ω la velocidad angular, t el tiempo, y φ la fase. Al método se le pasan como parámetros la longitud L del péndulo, la amplitud, el tiempo y la fase. La relación entre la longitud del péndulo y la velocidad angular ω es:

$$\omega = \sqrt{\frac{g}{L}}$$

siendo $g=9.8 \text{ m/s}^2$ la gravedad.

- 3) Indicar justificándolo, si es posible declarar directamente dentro de un método en Java los siguientes elementos. Se valorará la precisión y brevedad de la respuesta.
 - otros métodos
 - variables
 - constantes
 - atributos estáticos
- 4) Indicar cuáles son las ventajas de tener un sistema de ficheros jerárquico en un sistema operativo. Se valorará la precisión y brevedad de la respuesta.
- 5) Se dispone de la clase `EstacionMeteo` cuya interfaz se muestra en la página siguiente. representa un módulo software capaz de conectarse por SMS con una estación meteorológica remota, y obtener información. Escribir un fragmento de programa que cree dos objetos de esta clase, con los números de móvil que se indican en la tabla, y

luego ponga en pantalla para cada una de las dos estaciones meteorológicas su nombre, temperatura actual, y precipitación acumulada en las últimas 12 horas.

Nombre estación	Número de móvil GSM
Aeropuerto	607556100
Peña Cabarga	607556101

Class EstacionMeteo

```
java.lang.Object  
└ EstacionMeteo
```

```
public class EstacionMeteo  
extends java.lang.Object
```

Estacion Meteorologica accesible por SMS

Constructor Summary

[EstacionMeteo](#)(java.lang.String numMovil)
Constructor al que se le pasa el numero de movil

Method Summary

double	precipitacionAcumulada (int numHoras) retorna la precipitacion caida en las ultimas numHoras horas (l/m2)
double	tempActual () retorna la temperatura actual (grados centigrados)

Examen de Fundamentos de Computadores y Lenguajes

Examen Parcial. Febrero 2005

Segunda parte: Problema (5 puntos)

Se desea escribir un programa para hacer cálculos sobre el movimiento de una partícula cargada de carga q y masa en reposo M en un acelerador lineal con un campo eléctrico constante de intensidad E . Si utilizamos las ecuaciones de la mecánica relativista, la velocidad v de la partícula en función del tiempo t es

$$v = \frac{(qEt/Mc)}{\sqrt{1 + (qEt/Mc)^2}}c$$

siendo c la velocidad de la luz. En cambio, si usamos las ecuaciones de la dinámica no relativista, la velocidad de la partícula en función del tiempo es

$$v = \frac{qEt}{M}$$

El programa que se pide tendrá dos clases. En la primera clase, llamada `Particula`, se almacenan los datos de la partícula y se definen los métodos que permiten el cálculo de las velocidades. La clase obedece al siguiente diagrama de clase:

Particula	
atributos	<code>double carga</code> <code>double masa</code>
métodos	<code>Particula (double cargaParticula, double masaParticula)</code> <code>double velRelativista (double campo, double tiempo)</code> <code>double velNoRelativista (double campo, double tiempo)</code>

- Los atributos deben ser privados; sus unidades serán Culombios para la carga y Kg para la masa; se creará también una constante para almacenar la velocidad de la luz (m/s)
- El constructor recibe en los parámetros la carga (Culombios) y la masa de la partícula (Kg), y los copia en los respectivos atributos.
- Los dos métodos últimos calculan y retornan la velocidad de la partícula, respectivamente con cada una de las fórmulas de arriba. Se les pasa el campo eléctrico en (V/m) y el tiempo en segundos. La velocidad se obtiene en m/s.

La segunda clase, llamada `AceleradorLineal`, contiene el programa principal. Las acciones que debe realizar son las siguientes:

- Lee de teclado la carga y la masa de la partícula y crea usando los datos leídos un objeto de la clase `Particula`

- Lee de teclado el campo eléctrico y el tiempo y calcula las velocidades relativista y no relativista, almacenándolas en sendas variables
- Si la velocidad relativista es menor que el 99% de la no relativista pone en pantalla el mensaje "Debe usarse la velocidad relativista", e indica el valor de la velocidad relativista
- En caso contrario, puesto que las velocidades son muy parecidas, pone en pantalla el mensaje "Puede usarse la velocidad no relativista" y pone a continuación el valor de la velocidad no relativista.

Se valorará la clase `Particula` en dos puntos, y la clase `AceleradorLineal` en tres puntos.