

Soluciones del Examen de Fundamentos de Computadores y Lenguajes

Examen Parcial. Febrero 2004

Cuestiones (5 cuestiones, 5 puntos en total)

Cuestión 1

- 1) Crear una clase para guardar en tres atributos *públicos* el nombre, número y peso atómico de un átomo. Asimismo, crear un constructor al que se le pasen como parámetros tres datos que serán los valores iniciales de los atributos mencionados.

```
public class Atomo
{
    public String nombre;
    public int num;
    public double peso;

    public Atomo(String nombreAtomo, int numAtomo, double pesoAtomo)
    {
        nombre=nombreAtomo;
        num=numAtomo;
        peso=pesoAtomo;
    }
}
```

Cuestión 2

- 2) Se dispone de una clase que almacena en los siguientes atributos la medida de la velocidad de caída en el aire de una gota de un líquido y la densidad de ese líquido:

```
private double velCaida; // m/s
private double densidadLiquido; // Kg/m3
```

Crear un método sin parámetros que se añadiría a esa clase y que retorne el radio de la gota, que se puede calcular con la siguiente expresión:

$$r_{gota} = \sqrt{\frac{9\eta v_{cae}}{2\rho g}}$$

donde v_{cae} es la velocidad de caída, ρ es la densidad del líquido, g es la gravedad ($9.8m/s^2$) y η es un coeficiente de rozamiento que para temperatura ambiente vale $1.84 \cdot 10^{-5} Kg \cdot m^{-1} \cdot s^{-1}$

```
public double radio ()
{
    final double eta=1.84e-5; // kg/(m*s)
    final double g=9.8; // m/s2
    return Math.sqrt(9.0*eta*velCaida/(2.0*densidadLiquido*g));
}
```

Cuestión 3

- 3) Se dispone de una clase que permite almacenar la masa, posición y velocidad actual de una partícula. A continuación se muestran las cabeceras de sus métodos:

```
public class Particula {
    /**
     * Constructor, al que se le pasa la masa de la partícula en Kg */
    public Particula(double masa)...

    /**
     * Poner la posición. Coordenadas en metros
     */
    public void ponPosicion(double x, double y, double z)...

    /**
     * Poner la velocidad. Coordenadas en metros/s
     */
    public void ponVelocidad(double vx, double vy, double vz)...

    /**
     * Retorna la coord. x de la posición tras el tiempo indicado */
    public double calculaPosX(double tiempo) ...

    /**
     * Retorna la coord. y de la posición tras el tiempo indicado */
    public double calculaPosY(double tiempo) ...

    /**
     * Retorna la coord. z de la posición tras el tiempo indicado */
    public double calculaPosZ(double tiempo) ...
}

```

Se pide hacer un programa principal que cree un objeto de esa clase con masa 10^{-5} Kg. Luego, que establezca la posición a las coordenadas (3.0,-2.0, 10^{-2}) usando el método `ponPosicion()`. Finalmente, que ponga la velocidad a (0.0,0.0,2.3) usando el método `ponVelocidad()`.

```
public class CalculosParticula
{
    public static void main(String args[]){
        Particula p=new Particula(1.0e-5);
        p.ponPosicion(3.0,-2.0, 1.0e-2);
        p.ponVelocidad(0.0,0.0,2.3);
    }
}

```

Cuestión 4

- 4) Añadir al programa principal anterior instrucciones para mostrar en pantalla (System.out) las coordenadas de la posición del punto dentro de 3 segundos, mediante los métodos `calculaPos...()`. El formato para mostrar los datos será como en este ejemplo:

Coordenadas a los 3 segundos: x=3.0 y=-2.0 z=4.33

```
double px=p.calculaPosX(3.0);
double py=p.calculaPosY(3.0);
double pz=p.calculaPosZ(3.0);
System.out.println("Coordenadas a los 3 segundos: x="+
    px+" y="+py+" z="+pz);

```

Cuestión 5

- 5) Indicar cuál o cuáles son las órdenes necesarias en un sistema Linux para que, si nuestro directorio de trabajo es `/home/pcm/misdatos`, podamos borrar todos los ficheros que acaben con las extensiones `".jpg"` y `".gif"` (comillas excluidas) y que estén situados en el directorio `/home/pcm/antiguo`

Solución 1:

```
rm /home/pcm/antiguo/*.jpg
rm /home/pcm/antiguo/*.gif
```

Solución 2:

```
rm ../antiguo/*.jpg
rm ../antiguo/*.gif
```

Solución 3:

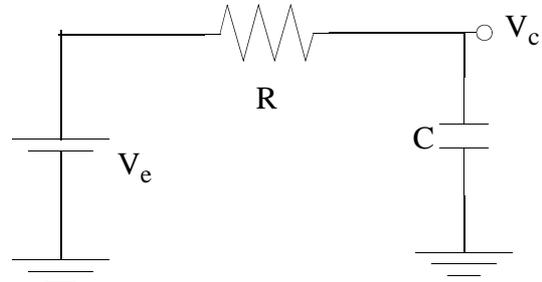
```
cd ../antiguo
rm *.jpg
rm *.gif
cd ../misdatos
```

Soluciones del Examen de Fundamentos de Computadores y Lenguajes

Examen Parcial. Febrero 2004

Problema (5 puntos)

Se desea hacer un programa para calcular el estado de un circuito formado por un condensador y una resistencia, como se muestra en la figura. El circuito se conecta a una fuente de corriente continua de tensión V_e . El condensador tiene una tensión inicial igual a V_i . Deseamos calcular el valor de V_c en función del tiempo.



El programa que se pide realizar consta de dos clases: la clase `Circuito` que guarda los datos del circuito y tiene métodos para obtener los valores eléctricos, y la clase `OperaCircuito` que contiene el método principal (`main`).

La clase `Circuito` está compuesta por:

- *atributos*: la capacidad del condensador en faradios, C , la resistencia en ohmios, R , el voltaje de la fuente en voltios, V_e , y el voltaje inicial del condensador en voltios, V_i .
- *métodos*:
 - un constructor al que se le pasan como parámetros los valores iniciales de los atributos, para copiarlos.
 - un método llamado `tensionCondensador` con un parámetro del tipo `double` que representa el tiempo en segundos (t) y que retorna el valor de la tensión del condensador, V_c , según la siguiente expresión:

$$V_c = V_i + (V_e - V_i)(1 - e^{-t/(RC)})$$

La clase `OperaCircuito` tiene el método `main` que hace lo siguiente:

- Lee de teclado (mediante un objeto de la clase `Lectura`) los datos necesarios para crear el circuito (C , R , V_e , y V_i).
- Crea un objeto de la clase `Circuito` usando los datos leídos del teclado.
- Calcula y almacena en tres variables el valor de la tensión del condensador al cabo de 1ms, 2ms, y 10ms usando el método `tensionCondensador()`.
- Muestra en una ventana de la clase `Escritura` los tres resultados obtenidos.

Solución

```
public class Circuito
{
    private double c; // capacidad en faradios
    private double r; // resistencia en ohmios
    private double ve; // voltaje de la fuente, en voltios
    private double vi; // Voltaje inicial del condensador, voltios
}
```

```

public Circuito(double c, double r, double ve, double vi)
{
    this.c=c;
    this.r=r;
    this.ve=ve;
    this.vi=vi;
}

public double tensionCondensador(double t)
{
    return vi+(ve-vi)*(1-Math.exp(-t/(r*c)));
}
}

import fundamentos.*;
public class OperaCircuito
{
    public static void main(String args[]){
        Lectura lec=new Lectura ("Datos Circuito");
        double c,r,ve,vi;
        double v1,v2,v3;

        lec.creaEntrada("C (faradios)",0.0);
        lec.creaEntrada("R (ohmios)",0.0);
        lec.creaEntrada("Ve (voltios)",0.0);
        lec.creaEntrada("Vi (coltios)",0.0);
        lec.espera();
        c=lec.leeDouble("C (faradios)");
        r=lec.leeDouble("R (ohmios)");
        ve=lec.leeDouble("Ve (voltios)");
        vi=lec.leeDouble("Vi (coltios)");

        Circuito circ = new Circuito(c,r,ve,vi);
        v1=circ.tensionCondensador(0.001);
        v2=circ.tensionCondensador(0.002);
        v3=circ.tensionCondensador(0.010);

        Escritura res=new Escritura("Valores de tension");
        res.insertaValor("tension al cabo de 1ms:",v1);
        res.insertaValor("tension al cabo de 2ms:",v2);
        res.insertaValor("tension al cabo de 10ms:",v3);
        res.espera();
    }
}
}

```