

Examen de Fundamentos de Computadores y Lenguajes

Examen Parcial. Febrero 2003

Cuestiones (5 cuestiones, 5 puntos en total)

- 1) Escribir un fragmento de programa Java que, utilizando la instrucción `switch`, y dada la variable entera `mes` que contiene el número de un mes (entre 1 y 12) almacene en la variable entera `dias` (ya declarada) el número de días de ese mes (que son, respectivamente para enero, febrero,...: 31,28,31,30,31,30,31,31,30,31,30,31).

```
switch (mes) {
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12:
        dias=31;
        break;
    case 4:
    case 6:
    case 9:
    case 11:
        dias=31;
        break;
    case 2:
        dias=28;
}
```

Fallos frecuentes:

- poner el número de días en la pantalla en lugar de en la variable `dias`
- olvidarse el `break`

- 2) Escribir en Java la siguiente expresión, suponiendo que las variables `x`, `a1`, `a2`, `a3`, e `y` están declaradas con el tipo `double`

$$x = \frac{\sin(a1) \cdot \cos\left(y^2 - \frac{a2}{2}\right)}{\sqrt{\tan(y + a3^{1/3})}}$$

```
x=Math.sin(a1)*Math.cos(y*y-a2/2)/
    Math.sqrt(Math.tan(y+Math.pow(a3,1.0/3.0)));
```

Fallos frecuentes:

- poner `1/3` en lugar de `1.0/3.0` (`1/3` es cero)
- confundir la variable `a2` con `a*2`
- usar paréntesis cuadrados y llaves en lugar de sólo paréntesis redondos

- 3) ¿Se puede asignar en Java un número real a una variable entera? ¿Y un número entero a una variable real? Razonar la respuesta.

El lenguaje Java tiene tipificación estricta (es decir, no permite mezclar datos de diferentes tipos en las expresiones), pero permite conversiones implícitas de números siempre que no haya posibilidad de pérdida de precisión.

Esto quiere decir que se puede asignar un número entero a una variable real, pues no hay pérdida de precisión. En cambio al revés no, pues puede que el número real sea muy grande y no quepa en el número entero, o que se pierda la parte fraccionaria.

- 4) Reescribe el siguiente fragmento de programa Java poniendo correctamente los sangrados o márgenes de texto:

```
public class Prueba {
    public static void main (String[] args) {
        int i=0; int k,l;
        int j=0;
        if (i*4>7) {
            j=i*4+7;
            if (j<2)
                l=45;
        } else {
            if (j<4) k=j-i;
            switch (j) {
                case 1:
                    l=34;
                    break;
                case 2:
                    {
                        l=35;
                    }
                    break;
            }
        }
    }
}
```

- 5) Suponer que disponemos de dos ficheros denominados `clase1.java` y `clase2.java` situados respectivamente en los directorios `/home/pepe/jbproject/ejemp/src` y `/home/pepe/otros-proyectos/proyecto33/src` de un computador Linux. El directorio de trabajo es `/home/pepe`. Escribir las instrucciones del intérprete de órdenes Linux necesarias para cambiar el nombre de estos ficheros a `Clase1.java` y `Clase2.java` respectivamente (es decir, con la primera letra en mayúscula).

```
mv /home/pepe/jbproject/ejemp/src/clase1.java /home/pepe/jbproject/
ejemp/src/Clase1.java
mv /home/pepe/otros-proyectos/proyecto33/src/clase2.java /home/
pepe/otros-proyectos/proyecto33/src/Clase2.java
```

otra solución:

```
mv jbproject/ejemp/src/clase1.java jbproject/ejemp/src/Clase1.java
mv otros-proyectos/proyecto33/src/clase2.java otros-proyectos/
proyecto33/src/Clase2.java
```

otra más:

```
cd jbproject/ejemp/src/
mv clase1.java Clase1.java
cd ../../../../otros-proyectos/proyecto33/src/
mv clase2.java Clase2.java
```

Examen de Fundamentos de Computadores y Lenguajes

Examen Parcial. Febrero 2003

Problema (5 puntos)

Se desea hacer un programa para calcular la distancia de un punto a una recta en el plano. El programa operará de manera que la ecuación de la recta se pueda introducir de dos maneras:

Ecuación general:

$$\frac{x - x_0}{a} = \frac{y - y_0}{b}$$

Ecuación 2:

$$Ax + By + C = 0$$

Se pide un programa Java que debe hacer lo siguiente:

- Leer de teclado las coordenadas x e y del punto (dos números reales), así como el tipo de ecuación (un número entero: 1=ecuación general; 2=ecuación 2)
- Si se ha elegido la ecuación general, leer de teclado los coeficientes x_0 , y_0 , a y b
- Si se ha elegido la ecuación 2, leer de teclado los coeficientes A , B , y C , y luego obtener la ecuación general de la recta según la ecuación:

$$A = b \quad B = -a \quad C = a \cdot y_0 - b \cdot x_0$$

- Calcular la distancia entre el punto y la recta usando la ecuación:

$$d = \frac{|Ax + By + C|}{\sqrt{A^2 + B^2}}$$

- Mostrar en pantalla la distancia obtenida.

Solución:

```
import fundamentos.*;

public class PuntoRecta {

    public static void main (String[] args) {

        final int ecuacionContinua=1, ecuacionImplicita=2;

        Lectura pant=new Lectura("Distancia Recta-Plano");
        double dist;
```

```

double x,y;          // punto
double a,b,c;       // ecuacion implicita
double ac,bc,x0,y0; // ecuacion continua
int tipoEcuacion;

// lectura del punto y tipo de ecuacion
pant.creaEntrada("Coordenada X del punto",0.0);
pant.creaEntrada("Coordenada Y del punto",0.0);
pant.creaEntrada("Tipo ec. recta: 1=continua, 2=implicita",1);
pant.espera();
x=pant.leeDouble("Coordenada X del punto");
y=pant.leeDouble("Coordenada Y del punto");
tipoEcuacion=
    pant.leeInt("Tipo ec. recta: 1=continua, 2=implicita");

//lectura de la ecuacion de la recta
switch (tipoEcuacion) {
case ecuacionContinua:
    pant.creaEntrada("coef x0 de la recta",0.0);
    pant.creaEntrada("coef y0 de la recta",0.0);
    pant.creaEntrada("coef a de la recta",0.0);
    pant.creaEntrada("coef b de la recta",0.0);
    pant.espera();
    x0=pant.leeDouble("coef x0 de la recta");
    y0=pant.leeDouble("coef y0 de la recta");
    ac=pant.leeDouble("coef a de la recta");
    bc=pant.leeDouble("coef b de la recta");
    // convertir a ecuacion implicita
    a=bc;
    b=-ac;
    c=ac*y0-bc*x0;
    break;
case ecuacionImplicita:
    pant.creaEntrada("coef a de la recta",0.0);
    pant.creaEntrada("coef b de la recta",0.0);
    pant.creaEntrada("coef c de la recta",0.0);
    pant.espera();
    a=pant.leeDouble("coef a de la recta");
    b=pant.leeDouble("coef b de la recta");
    c=pant.leeDouble("coef c de la recta");
    break;
default:
    pant.println("Tipo erroneo");
    return;
}

// Calcular la distancia y mostrar el resultado
dist=Math.abs(a*x+b*y+c)/Math.sqrt(a*a+b*b);

pant.println("Distancia="+dist);
}
}

```