

Nuevas Tecnologías de la Programación

11 de Febrero de 2010

1. (1 punto) ¿Por qué se dice que el sistema X-Window está basado en una arquitectura cliente-servidor?.
2. (1 punto) Supongamos que un programa hecho en Xlib gestiona los eventos `Expose` de una ventana, repintando todos los contenidos de la ventana cada vez que ocurre tal evento. Explica cómo mejorarías la eficiencia del tratamiento del evento `Expose` usando el campo `count` de tal evento.
3. (1 punto) Explica qué es el *estado* (data state) de un objeto de una clase interna no estática de Java. Explica si es posible que varios objetos de una clase interna no estática, puedan compartir el mismo objeto de la clase externa
4. (3.5 puntos) Escribe los pasos más importantes de una aplicación (haciendo uso de la biblioteca Qt), que permita al usuario dibujar círculos dentro de un `QWidget`. El centro de cada círculo se introducirá pinchando con el ratón en el `QWidget`. Los círculos tendrán siempre un radio de 10 pixels.
5. (3.5 puntos) Escribe un programa en Java que muestre una ventana (`Frame` o `JFrame`) con un botón (`Button` o `JButton`) que permita salir del programa cuando se pulse. Además el frame contendrá un panel (clase `Canvas` o `JPanel`) en el que debe aparecer dibujado un pequeño cuadrado de 20×20 , inicialmente situado en la esquina superior izquierda del anterior panel. Cuando comience el programa, el cuadrado empezará a moverse de 10 en 10 pixel cada 100 milisegundos hacia la derecha. El usuario puede cambiar la dirección de movimiento con las teclas de cursor del teclado. Para el movimiento del cuadrado debe crearse una hebra (`Thread`) que se encarga de mover el cuadrado cada 100 milisegundos en la dirección actualmente seleccionada. Para obtener la hora actual podemos usar:

```
Date date=new Date();
Calendar cal=Calendar.getInstance();
cal.setTime(date);
int segundos=cal.get(Calendar.SECOND);
int minutos=cal.get(Calendar.MINUTE);
int horas=cal.get(Calendar.HOUR);
```

El evento de pulsación en el botón se controla con el interfaz `ActionListener` implementando el método:

```
void actionPerformed(ActionEvent event).
```

Nuevas Tecnologías de la Programación

11 de Febrero de 2010

1. (1 punto) ¿Por qué se dice que el sistema X-Window está basado en una arquitectura cliente-servidor?.
2. (1 punto) Supongamos que un programa hecho en Xlib gestiona los eventos `Expose` de una ventana, repintando todos los contenidos de la ventana cada vez que ocurre tal evento. Explica cómo mejorarías la eficiencia del tratamiento del evento `Expose` usando el campo `count` de tal evento.
3. (1 punto) Explica qué es el *estado* (data state) de un objeto de una clase interna no estática de Java. Explica si es posible que varios objetos de una clase interna no estática, puedan compartir el mismo objeto de la clase externa
4. (3.5 puntos) Escribe los pasos más importantes de una aplicación (haciendo uso de la biblioteca Qt), que permita al usuario dibujar círculos dentro de un `QWidget`. El centro de cada círculo se introducirá pinchando con el ratón en el `QWidget`. Los círculos tendrán siempre un radio de 10 pixels.
5. (3.5 puntos) Escribe un programa en Java que muestre una ventana (`Frame` o `JFrame`) con un botón (`Button` o `JButton`) que permita salir del programa cuando se pulse. Además el frame contendrá un panel (clase `Canvas` o `JPanel`) en el que debe aparecer dibujado un pequeño cuadrado de 20×20 , inicialmente situado en la esquina superior izquierda del anterior panel. Cuando comience el programa, el cuadrado empezará a moverse de 10 en 10 pixel cada 100 milisegundos hacia la derecha. El usuario puede cambiar la dirección de movimiento con las teclas de cursor del teclado. Para el movimiento del cuadrado debe crearse una hebra (`Thread`) que se encarga de mover el cuadrado cada 100 milisegundos en la dirección actualmente seleccionada. Para obtener la hora actual podemos usar:

```
Date date=new Date();
Calendar cal=Calendar.getInstance();
cal.setTime(date);
int segundos=cal.get(Calendar.SECOND);
int minutos=cal.get(Calendar.MINUTE);
int horas=cal.get(Calendar.HOUR);
```

El evento de pulsación en el botón se controla con el interfaz `ActionListener` implementando el método:

```
void actionPerformed(ActionEvent event).
```