

Nuevas Tecnologías de la Programación

29 de Enero de 2009

1. (1 punto) Explica y compara brevemente qué mecanismo (struct o clase) tienen Xlib, Qt y Java para definir las características que usarán las primitivas gráficas (para dibujar líneas, círculos, etc).
2. (1 punto) Explica brevemente cuáles son los pasos más importantes en un programa Xlib.
3. (1 punto) Supongamos que tenemos un programa en Java con una clase A y otra clase B que es subclase de A. Sea un método $met(int\ u, int\ v)$ que está definido en la clase A y sobrescrito en la clase B y un método $met2(int\ u)$ definido únicamente en la clase B. Explica lo que ocurre en las siguientes dos situaciones:

a) `A a;`
`B b=new B();`
`a=b;`
`a.met(10,20);`

b) `A a;`
`B b=new B();`
`a=b;`
`a.met2(10);`

4. (3.5 puntos) Escribe un programa en Qt que muestre una ventana que permita al usuario dibujar rectángulos con el uso del ratón, de forma que no desaparezcan cuando se produzca un evento `Expose`. Los rectángulos se dibujan pinchando con un botón del ratón para indicar la esquina superior izquierda y volviendo a pinchar para indicar la esquina inferior derecha.
5. (3.5 puntos) Escribe un programa en Java que muestre una ventana (**Frame**) con un **Canvas** que permita medir la velocidad a la que movemos el ratón por el **Canvas**. La velocidad comienza a medirse cuando pulsamos (evento `mousePressed` del interfaz `MouseListener`) el botón izquierdo del ratón y termina de medirse cuando lo soltamos (evento `mouseReleased` del interfaz `MouseListener`). En ese momento habría que mostrar la velocidad calculada en un objeto **Label**. La velocidad se calcula como la distancia recorrida dividida por el tiempo transcurrido. Para medir la distancia podemos controlar el evento `mouseDragged` del interfaz `MouseMotionListener` que nos indica que hemos movido el ratón con un botón pulsado. Cada vez que se produzca este evento sumariamos la distancia parcial recorrida, a la distancia total. Para medir el tiempo transcurrido podemos hacer uso del método `long getTime()` (de la clase `java.util.Date`) que nos devuelve el número de milisegundos transcurridos desde el 1 de Enero de 1970 para el objeto `Date` asociado. Se usaría de la siguiente forma:

```
Date date=new Date();  
long segTranscurridos=date.getTime();
```

Nuevas Tecnologías de la Programación

29 de Enero de 2009

1. (1 punto) Explica y compara brevemente qué mecanismo (struct o clase) tienen Xlib, Qt y Java para definir las características que usarán las primitivas gráficas (para dibujar líneas, círculos, etc).
2. (1 punto) Explica brevemente cuáles son los pasos más importantes en un programa Xlib.
3. (1 punto) Supongamos que tenemos un programa en Java con una clase A y otra clase B que es subclase de A. Sea un método $met(int u, int v)$ que está definido en la clase A y sobrescrito en la clase B y un método $met2(int u)$ definido únicamente en la clase B. Explica lo que ocurre en las siguientes dos situaciones:

a) `A a;`
`B b=new B();`
`a=b;`
`a.met(10,20);`

b) `A a;`
`B b=new B();`
`a=b;`
`a.met2(10);`

4. (3.5 puntos) Escribe un programa en Qt que muestre una ventana que permita al usuario dibujar rectángulos con el uso del ratón, de forma que no desaparezcan cuando se produzca un evento `Expose`. Los rectángulos se dibujan pinchando con un botón del ratón para indicar la esquina superior izquierda y volviendo a pinchar para indicar la esquina inferior derecha.
5. (3.5 puntos) Escribe un programa en Java que muestre una ventana (**Frame**) con un **Canvas** que permita medir la velocidad a la que movemos el ratón por el **Canvas**. La velocidad comienza a medirse cuando pulsamos (evento `mousePressed` del interfaz `MouseListener`) el botón izquierdo del ratón y termina de medirse cuando lo soltamos (evento `mouseReleased` del interfaz `MouseListener`). En ese momento habría que mostrar la velocidad calculada en un objeto **Label**. La velocidad se calcula como la distancia recorrida dividida por el tiempo transcurrido. Para medir la distancia podemos controlar el evento `mouseDragged` del interfaz `MouseMotionListener` que nos indica que hemos movido el ratón con un botón pulsado. Cada vez que se produzca este evento sumariamos la distancia parcial recorrida, a la distancia total. Para medir el tiempo transcurrido podemos hacer uso del método `long getTime()` (de la clase `java.util.Date`) que nos devuelve el número de milisegundos transcurridos desde el 1 de Enero de 1970 para el objeto `Date` asociado. Se usaría de la siguiente forma:

```
Date date=new Date();  
long segTranscurridos=date.getTime();
```