

I. Introducción a la programación orientada a objetos y al lenguaje JAVA



Lectura de Datos

Dentro de un applet es sencillo tomar algún dato del usuario para realizar algún cálculo.

Es difícil explicar detalladamente los diferentes objetos en este momento que requieres para tomar un dato de la ventana del applet, cada objeto utilizado será explicado detalladamente en unidades posteriores, pero es importante que ya empieces a tomar datos del usuario, es por eso que te daremos los objetos más simples en este momento.

Esto se hace a través de crear un objeto de la clase `TextField`, el cual sirve para que el usuario teclee algún dato dentro del objeto, en la clase `TextField` existen diferentes métodos, es necesario utilizar el método `getText()` para tomar el dato del usuario y este método regresa un objeto de la clase `String` (cadena de caracteres), el cual se debe pasar a entero o número de punto flotante, utilizando la clase `Integer` o utilizando la clase `Double` (revisar ejemplo abajo).

Dentro de un Applet el método `paint` nos sirve para dibujar. En el método constructor (método con el mismo nombre de la clase) creamos objetos de la clase de elementos de interfaz gráfica, como `TextField`, `TextArea`, `Label`, `Button`, etc., y se añaden a la ventana del applet con el método `add`.

Cuando tomamos un dato del usuario, la manera que este le dice al applet que ya ha introducido el dato y quiere hacer el cálculo es dando un clic a un botón, el cual es creado dentro del applet por medio de la clase `Button`, para poder introducir las instrucciones para hacer algún cálculo cuando el usuario ya dio clic en el botón, debemos introducir éstas en el método `actionPerformed`, este método pertenece a la clase `ActionListener`, es entonces que la clase que estamos escribiendo que hereda de la clase `Applet` (clase `AppletDaCuadrado` en este caso), debe implementar la clase `ActionListener` para poder utilizar este método y dar instrucciones para cuando se da clic en el botón.

Es decir cuando queremos utilizar un Applet debemos heredar de la clase `Applet`, esto lo hacemos en la misma declaración de la clase:

```
public class AppletDaCuadrado extends Applet {
```

Habiendo definido previamente la instrucción `import java.applet.*;` el cual toma la clase `Applet` previamente definida por Java en la API.

Si un applet requiere utilizar elementos gráficos que el usuario utilizará para definir que requiere una acción como la de dar un clic en un botón, entonces ese Applet necesita implementar la interface `ActionListener`, la cual tiene métodos que ayudan a definir acciones para cuando el usuario da un clic en un botón, quedando entonces la declaración del applet como:

```
public class AppletDaCuadrado extends Applet implements ActionListener {
```

Es importante que no te sientas muy presionado por tener una serie de métodos nuevos, dentro de la clase Applet, pero pronto entre más ejercicios realices, más fácil te será entender estos nuevos métodos.

Presentaremos un applet sencillo en el que se toma un número del usuario y cuando el de un clic aparecerá el cuadrado de ese número dibujado:

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
```

```
// <applet width="200" height="200" code="AppletDaCuadrado"></applet>
```

```
public class AppletDaCuadrado extends Applet implements ActionListener {
    TextField t;
    Button b;
```

```
    //método constructor, donde se crean los objetos a dibujar en el applet
```

```
    public AppletDaCuadrado() {
        t = new TextField();    // se crea el objeto texto
        b = new Button("Calcula"); // se crea el boton para calcular
        add(t);    // se añade el texto a la pantalla
        add(b);    // se añade el botón a la pantalla
        b.addActionListener(this); // se le añade al boton la facilidad de
                                   // ser escuchado
    }
```

```
    //método paint en el cual se pone lo que se quiere dibujar
```

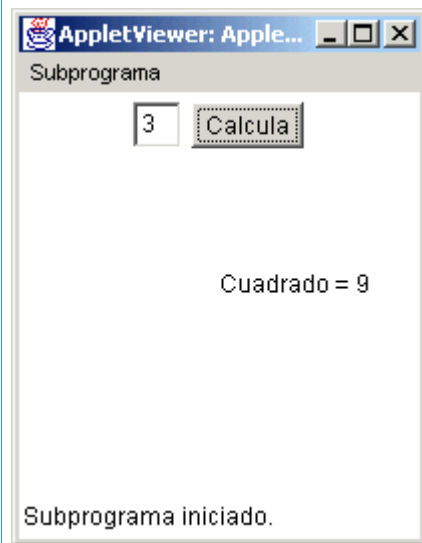
```
    public void paint(Graphics g) {
        int num = Integer.parseInt(t.getText()); // pasa texto al numero entero
        g.drawString("Cuadrado = "+num*num, 100, 100); //dibuja el cuadrado
    }
```

```
    // método para realizar las opciones al oprimir un botón
```

```
    public void actionPerformed(ActionEvent ae) {
        repaint(); // se le pide que se repinte la ventana
    }
```

```
}
```

La manera en la que se presenta este applet es la siguiente:



Analicemos un poco esto a detalle, la clase Applet tiene el método `paint` para dibujar, y el método `actionPerformed()` para llevar a cabo las instrucciones cuando alguna acción se de, en este caso se haga clic en un botón. Para poder hacer que un applet tenga elementos de interfaz gráfica, estos deben ser creados utilizando la clase deseada (`TextField`, `TextArea`, `Button`, etc) y deben ser añadidos a la aplicación a través del método `add` (para que aparezcan en la ventana del applet), todo esto se hace en el constructor del applet, de esa manera cuando se ejecute el applet, se crean los objetos de interfaz gráfica y se presentan en la ventana del applet.

La manera en la que se pueda hacer para que se ejecute alguna instrucción, con la acción de un botón, es primero añadirlo, pero además hacer que este elemento pueda ser escuchado, es decir saber cuando el usuario da un clic en el, al momento de estar ejecutando el applet, esto se hace a través del método `addActionListener(this)`, en la línea

```
b.addActionListener(this);
```

Aquí estamos diciendo que el botón debe ser escuchado en esta clase de aplicación, de esta manera cuando el usuario da un clic, entonces el control de la aplicación pasará al método `actionPerformed()` y se ejecutarán las instrucciones puestas en él.

```
repaint();
```

Esta instrucción lo que hace es hacer que se vuelva a ejecutar el método `paint()`, para que aparezca el mensaje `Cuadrado =` y el valor del cuadrado del número dado. Ahora porque tenemos que hacer esto, debido a que el método `paint()` solo se ejecuta cuando se mueve la ventana del applet, cuando se abre y se cierra, pero no en todo momento, entonces el método `repaint()` nos sirve para que se repinte lo que queremos y en este caso cuando el usuario da un clic en el botón.

Utilizamos el método `parseInt()` de la clase `Integer` para poder pasar el valor `String` (cadena de caracteres) del texto que tiene el número dado por el usuario, a un número entero que pueda servirnos posteriormente para hacer algún cálculo.

Ahora la realidad es que el método `paint` es para dibujar y no para dar respuestas a cálculos,

entonces lo mejor seria no utilizar el método paint() en este caso, sino utilizar otros elementos de interfaz gráfica para mostrar los cálculos, esto será en el siguiente applet:

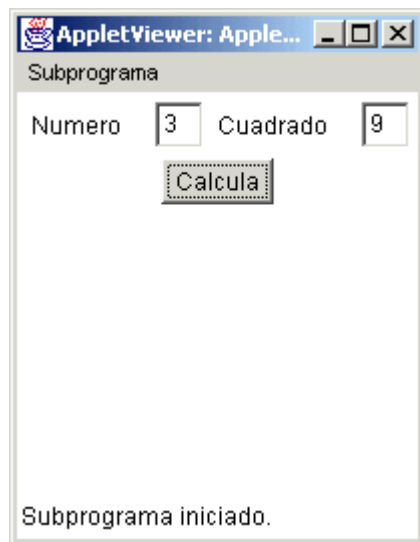
```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
import java.lang.Math;

// <applet width="200" height="200" code="AppletCuadrado"></applet>

public class AppletCuadrado extends Applet implements ActionListener {
    Label l1, l2;
    TextField t1, t2;
    Button b;

    public AppletCuadrado() {
        l1 = new Label("Numero"); // se crea una etiqueta
        t1 = new TextField();      // se crea un texto
        l2 = new Label("Cuadrado"); // se crea una etiqueta
        t2 = new TextField();      // se crea un texto
        b = new Button("Calcula"); // se crea un boton
        add(l1);                  // se añade la etiqueta a la ventana
        add(t1);                  // se añade un texto a la ventana
        add(l2);                  // se añade la segunda etiqueta a la ventana
        add(t2);                  // se añade el segundo texto a la ventana
        add(b);                   // se añade el boton a la ventana
        b.addActionListener(this); // el boton ahora tiene la facilidad de
                                   // ser escuchado
    }
    public void actionPerformed(ActionEvent ae) {
        // se toma el numero del texto y se pasa a la variable num
        double num = Double.parseDouble(t1.getText());
        // al texto 2 se le pone el valor del numero al cuadrado
        t2.setText(""+Math.pow(num,2.0));
    }
}
```

El cual al ejecutar muestra lo siguiente:



Utilizamos el método `parseDouble` de la clase `Double`, para pasar de un `String` a un valor primitivo de doble precisión (como lo hicimos en el ejemplo anterior con un entero), usamos un valor de doble precisión, porque ahora hicimos uso del método `pow()` de la clase `Math`, para hacer el cálculo del cuadrado. Ahora no hacemos uso del método `paint()`, y utilizamos el método `setText()` para poder pasar el valor calculado al objeto de la clase `TextField`, todo esto se hace dentro del método `actionPerformed()` y como no se está dibujando el resultado, no tenemos que hacer uso del método `paint()` o `repaint()`.

El uso del método `pow()` se hace tomando solo el nombre de la clase `Math`, haciendo `Math.pow(num, 2.0)`, esto se hace debido a que el método es estático, lo cual significa que el método se usa sin tener que crear un objeto de la clase `Math`, solo haciendo uso del nombre de la clase antes del método.

En esta clase también hicimos uso de objetos de la clase `Label`, para poder definir que es lo que estará en el texto de enseñanza.

Recuerda que estamos utilizando la primera etiqueta del applet para no tener que hacer un archivo `html` y poder utilizar el applet, pero cuando hagas un applet y lo quieras liberar, debes de escribir un documento `html` y poner la etiqueta del applet en el cuerpo del documento.