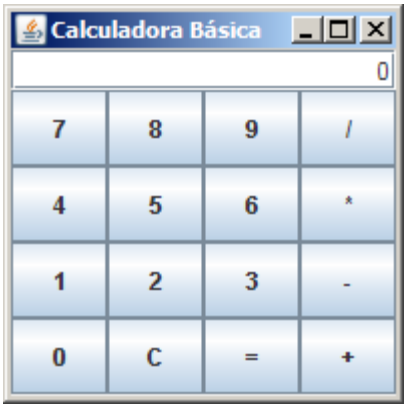
Programa **Calculadora**

Por cuestiones de espacio y legibilidad el código fuente de la calculadora en java se muestra en porciones.



Para crear la calculadora de la imágen anterior debemos tener en cuenta la interfaz y los eventos. Lo ideal es mantener el código que permite los cálculos en una clase separada, pero aqui todo se realizará en una única clase.

Para comenzar a trabajar con componentes gráficos y eventos debemos agregar al inicio del archivo de la clase las siguientes inclusiones:

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
```

Con esto importamos todas las clases awt, events y swing.

Estructura del programa

Nuestra clase se llamará Main y nuestro archivo main.class. Este es su esqueleto:

```
01 public class Main extends JFrame implements ActionListener {
02
03     private JTextField t;
04     private int a = 0,  b = 0;
05     private int eq = 0;
```

```
06      private char op = 0;
07
08      public Main() {
09          ...
10      }
11
12      public static void main(String[] args) {
13          new Main();
14      }
15
16      public void actionPerformed(ActionEvent e) {
17          ...
18      }
19  }
```

Como se puede ver, nuestra clase extiende JFrame e implementa la interface ActionListener que nos obliga a implementar el método actionPerformed (click en botón).
Se tienen atributos privados, el primero un cuadro de texto, los demas contadores y variables de control.
Luego tenemos el constructor de la clase, que nos servirá para crear la interfaz y asignar eventos a los controles correspondientes (lo veremos ahora mismo).
Luego tenemos un método estatico especial llamado main (en minúsculas) que nos permite ejecutar la calculadora.
Y por último un evento para capturar los clicks del usuario sobre los diferentes botones de la calculadora.

Creando la interfaz

Dentro del constructor podemos ver este código:

```
01 super("Calculadora Básica");
   StringlabelButtons[] =
02 {"7", "8", "9", "/", "4", "5", "6", "*", "1", "2","3", "-
   ", "0", "C", "=", "+"};
03 JPanelcp = (JPanel) this.getContentPane();
04 cp.setLayout(new BorderLayout());
05 JPanel p = new JPanel();
```

```
06 p.setLayout(new GridLayout(0, 4));
07 for (int i = 0; i <labelButtons.length; ++i) {
08         JButton button = new JButton(labelButtons[i]);
09         button.addActionListener(this);
10         p.add(button);
11 }
12 t = new JTextField();
13 t.setHorizontalAlignment(JTextField.RIGHT);
14 t.setText("0");
15 cp.add(t, BorderLayout.PAGE_START);
16 cp.add(p, BorderLayout.CENTER);
```

Primero asignamos el título de la ventana y creamos un vector con los caracteres de los botones. Si lo notaron tiene un orden algo extraño, esto se debe al algoritmo que se usa luego para crear los controles recorriendo el vector.

Posteriormente creamos un JPanel y le asignamos un layout tipo grid de 4 columnas, entonces al recorrer el vector vamos agregando a este panel objetos JButton creados con la etiqueta que obtenemos del item actual del vector y de paso ya le asignamos el controlador del evento (el mismo objeto, this, hace referencia a esta misma instancia de la clase Main).
Al salir del ciclo ya tenemos todos los botones, pero nos falta un poco para terminar el diseño. Creamos un cuadro de texto y le fijamos alineación de texto a la derecha (será donde se muestren los resultados entre otras cosas). Inicialmente le asignamos un texto igual a "0".
Al panel principal le colocamos el layoutBorderLayout, agregamos el cuadro de texto arriba y al centro el panel que contiene todos los botones generados anteriormente.

Capturando los eventos

Nuestra interfaz nos quedó muy bonita, pero no hace nada. Debemos darle funcionalidad y esto lo hacemos en el evento que captura los click del usuario sobre los diferentes botones de la interfaz.

```
01 public void actionPerformed(ActionEvent e) {
02     char c = ((JButton) e.getSource()).getText().charAt(0);
03     if (c >= '0' && c <= '9') {
```

```
04          .... Implementación
05      //Log
06      System.out.print(a);
07      System.out.print(" ");
08      System.out.print(b);
09      ....
10  }
```

La variable local **c** se asigna con el caracter que hace referencia al botón pulsado (ver el contenido del vector de etiquetas de botones que se declara en el constructor de la clase).
Con esa variable sabremos qué es lo que pulsó el usuario y consecuentemente deberemos trabajar con dicha orden.
Si es un dígito haremos una cosa, si es un signo de operatoria otra cosa. Si se trata del signo igual deberemos calcular e informar. Y si es el simbolo C (clear), simplemente resetearemos la calculadora.