GUI. Swing toolkit.

Using JOptionPanel, JFrame, JPanel

GUI

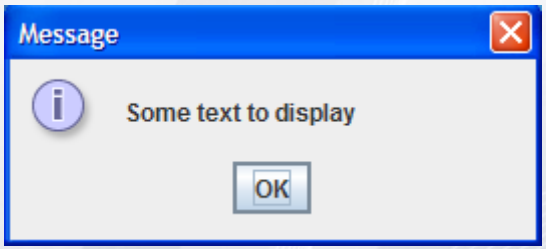
- What is Graphical user interface?
  - A type of user interface that allows users to interact with electronic devices with images rather than text commands.
  - A GUI represents the information and actions available to a user through graphical icons and visual indicators
- It's more efficient and easy for usage than the command line interface

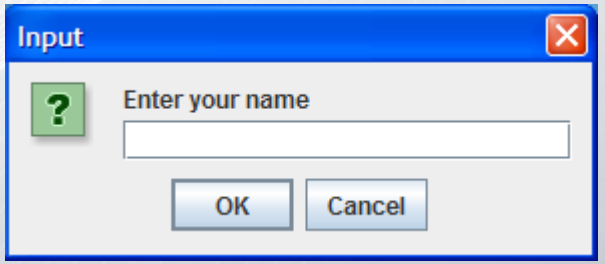
Swing

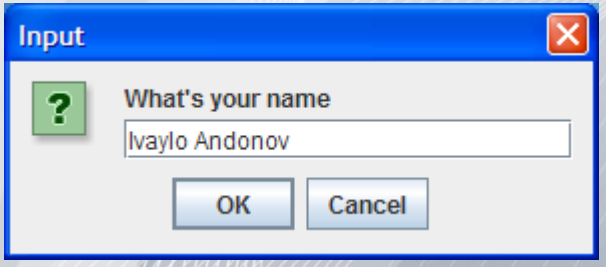
- Swing is the primary Java GUI widget toolkit.
- It allows applications to have a look and feel unrelated to the underlying platform
- It consists of many graphical components such as buttons, textfields, checkboxes ...
- It's better than its ancestor AWT
- All related classes are placed in javax.swing package

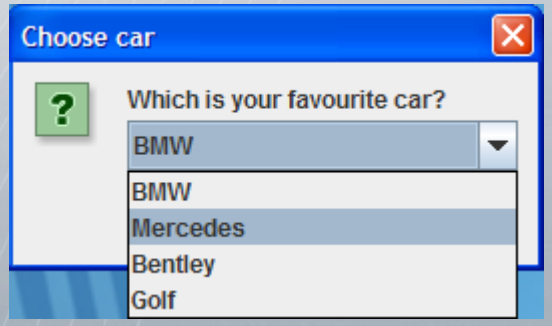
JOptionPane class

- It is part of the java swing library. It requires an import statement at the top of the program
- JOptionPane makes it easy to pop up a standard dialog box that prompts users for a value or informs them of something









Some of JOptionPane's methods

- `JOptionPane.showMessageDialog(...);`
- `JOptionPane.showConfirmDialog(...);`
- `JOptionPane.showInputDialog(...);`

- Almost all of the methods are static and we don't need an instance to use them
- Most of the methods are overloaded and do different things depending on the parameters

JOptionPane methods

```
public static void main(String[] args) {
    //output
    JOptionPane.showMessageDialog(null, "Some text to display");
    JOptionPane.showMessageDialog(null, "Some message", "Test",
                                  JOptionPane.ERROR_MESSAGE);
    JOptionPane.showMessageDialog(null, "Some message", "Test",
                                  JOptionPane.QUESTION_MESSAGE);

    //user input with yes no
    int choice = JOptionPane.showConfirmDialog(null, "Are you female?");
    System.out.println(choice);

    //user input
    String name = JOptionPane.showInputDialog("Enter your name");
    System.out.println(name);
    JOptionPane.showInputDialog("What's your name",
                                      "Enter your name here");

}
```

JOptionPane methods

```
public static void main(String[] args) {

    String[] cars = {"BMW", "Mercedes", "Bentley", "Golf"};
    JOptionPane.showInputDialog(null, "Which is your favourite car?",
"Choose car", JOptionPane.QUESTION_MESSAGE, null, cars, "BMW");

    Page[] pages = new Page[2];
    pages[0] = new Page("Page 1", "Text of the first page");
    pages[1] = new Page("Page 2", "Text of the second page");

    JOptionPane.showInputDialog(null, "Please, choose page?", "Choose    page",
JOptionPane.QUESTION_MESSAGE, null, pages, pages[1]);

}
```

The base for all GUI programs in Java

- To create graphical window in swing we need an instance of JFrame
- Also we must set its size
- And make it visible
- It's good practice to set its “default close operation“

- Every swing application consist of one JFrame and many panel (JPanel) added to it

The base for all GUI programs in Java

```
public class DrawingTest {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Drawing program");
        frame.setSize(1000, 700);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);

        // create panels and add it to the frame
    }
}
```

JPanel

- JPanel provide us with area for drawing or adding graphical components
- It's not good practice to add non container components (like buttons and text fields) directly to the frame

```
public static void main(String[] args) {
    JFrame frame = new JFrame("Drawing program");
    frame.setSize(1000, 700);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    JPanel panel = new JPanel();
    //add components to the panel

    //finally add the panel with the components in it to the frame
    frame.add(panel);
    frame.setVisible(true);
}
```

Drawing into a panel with Graphics

- The method `protected void paintComponent(Graphics g)` is responsible for drawing the panel when it's added to a frame
- To draw some elements to a panel (lines, rectangles, ovals...) we must override this method
- We'll use the instance of java.awt.Graphics to draw in the panel
- A graphics context's coordinate system is used to for drawing

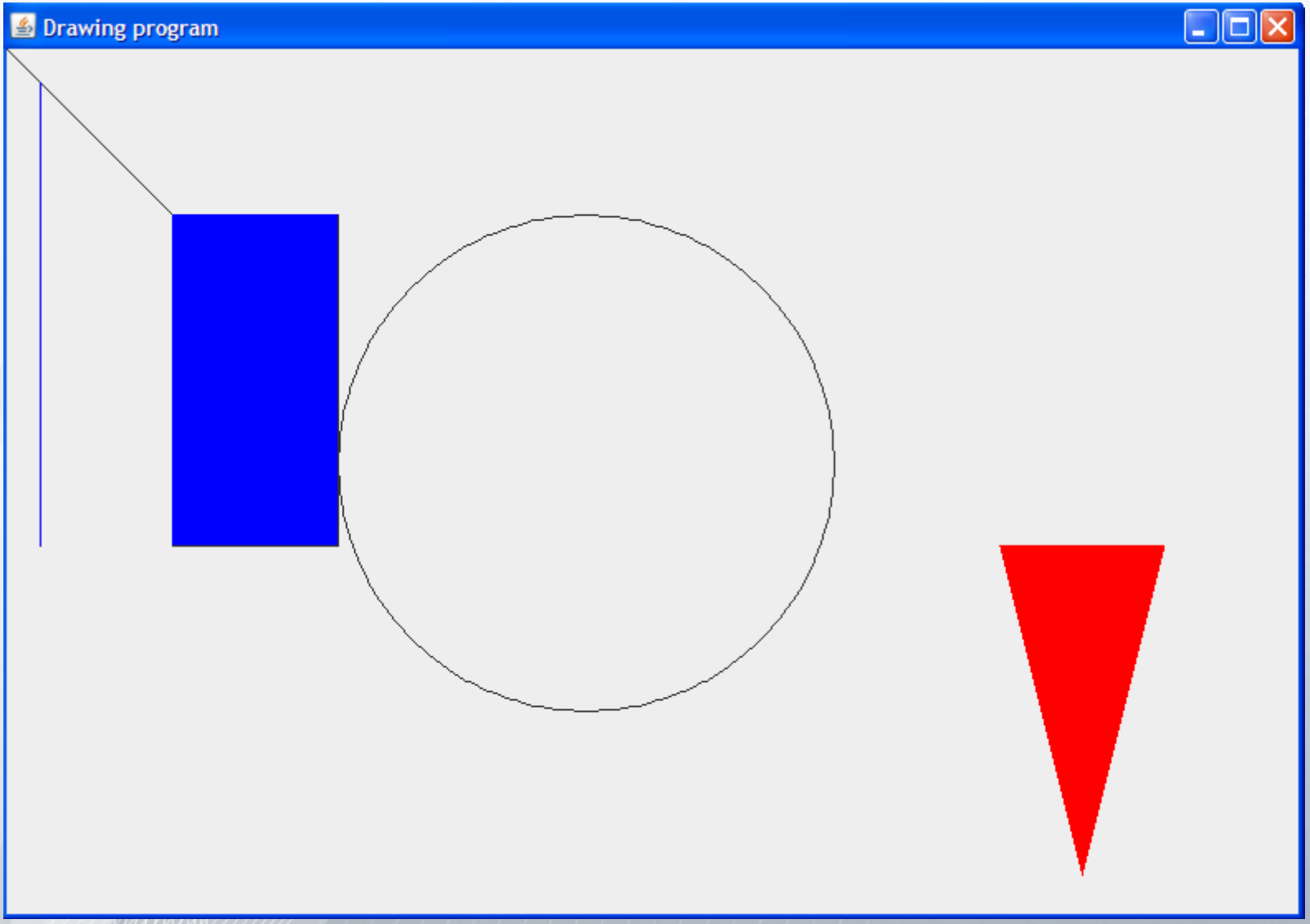
Drawing into a panel with Graphics

```
public class DrawingPanel extends JPanel{
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawLine(0, 0, 100, 100);
        g.drawRect(100, 100, 100, 200);
        g.drawOval(200, 100, 300, 300);

        //Color color = new Color(20, 33, 122);
        Color newColor = Color.BLUE;
        g.setColor(newColor);

        g.drawLine(20, 20, 20, 300);
        g.fillRect(100, 100, 100, 200);

        int[] x = {600, 700, 650};
        int[] y = {300, 300, 500};
        g.setColor(Color.RED);
        g.fillPolygon(x, y, 3);
    }
}
```

Drawing into a panel with Graphics

```
public class DrawingTest {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Drawing program");
        frame.setSize(1000, 700);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        DrawingPanel p = new DrawingPanel();
        frame.add(p);
        frame.setVisible(true);
    }
}
```

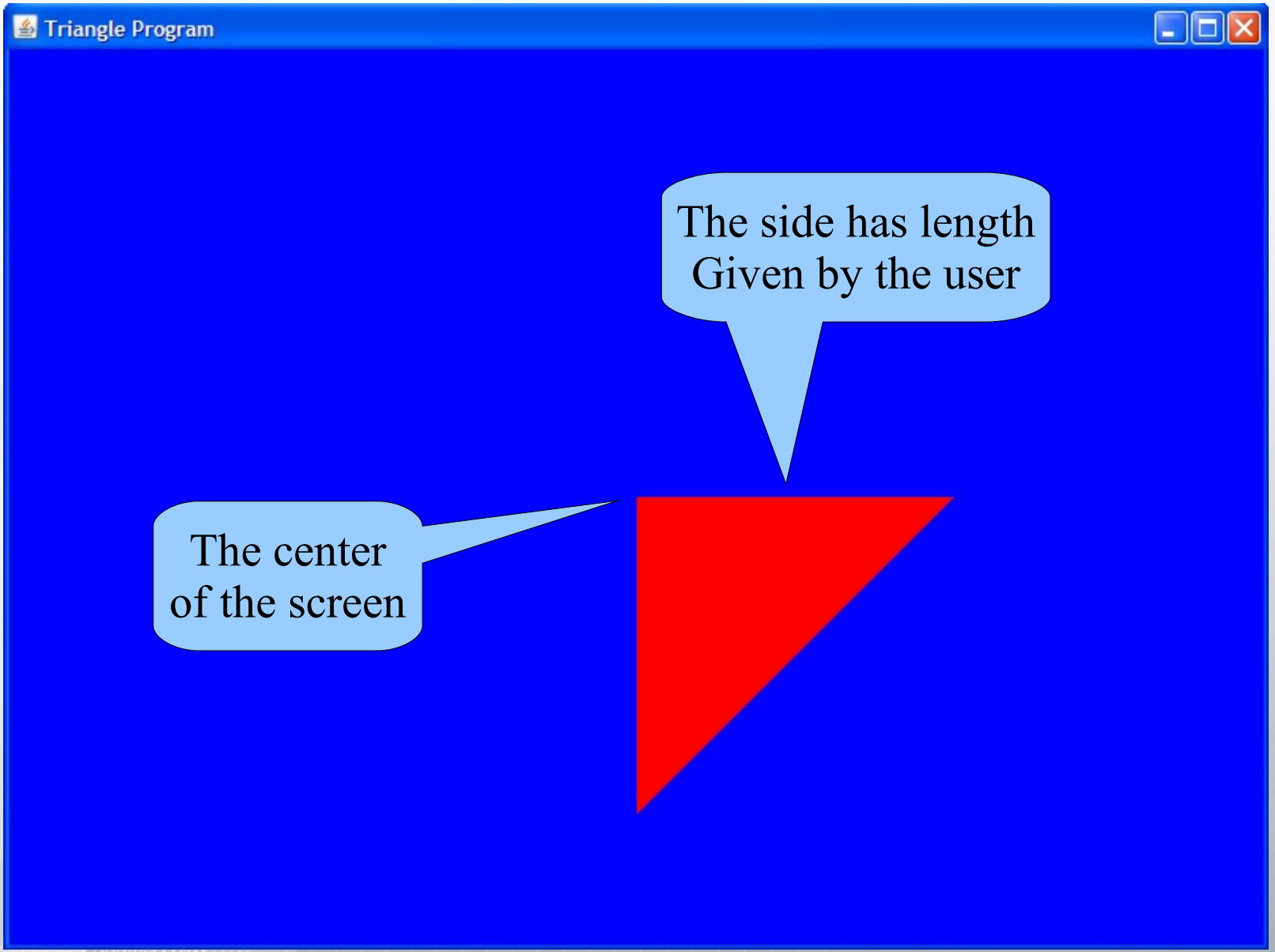
The result



Exercise

- Ask the user to enter size for the window (width and height)
- Ask the user to enter the length of the side of right triangle which will be drawn
- Draw right triangle with the given side with the right angle in the center of the screen
- Fill the triangle with red color and use blue for the background of the window
- If you want, you can ask the user to choose the colors

Exercise


Triangle Program
The side has length
Given by the user
The center
of the screen

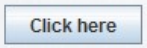
Exercise

```
public class TrianglePanel extends JPanel {
    private int side;

    public TrianglePanel(int side) {
        this.side = side;
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        int xCenter = this.getWidth() / 2;
        int yCenter = this.getHeight() / 2;

        this.setBackground(Color.BLUE);

        int[] x = {xCenter, xCenter + side, xCenter};
        int[] y = {yCenter, yCenter, yCenter + side};
        g.setColor(Color.RED);
        g.fillPolygon(x, y, 3);
    }
}
```

Exercise

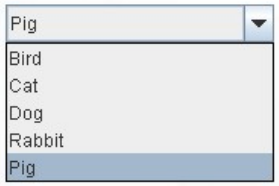
```
public class TrianglePanelTest {

    public static void main(String[] args) {
        String widthStr = JOptionPane.showInputDialog(null, "Enter width");
        int width = Integer.parseInt(widthStr);

        String heightStr = JOptionPane.showInputDialog(null, "Enter height");
        int height = Integer.parseInt(heightStr);

        String sideStr = JOptionPane.showInputDialog(null, "Enter side");
        int side = Integer.parseInt(sideStr);

        JFrame f = new JFrame("Triangle Program");
        f.setSize(width, height);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setVisible(true);

        TrianglePanel p = new TrianglePanel(side);
        f.add(p);
    }
}
```

Demonstrating some of the graphical components

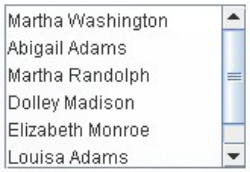


JButton

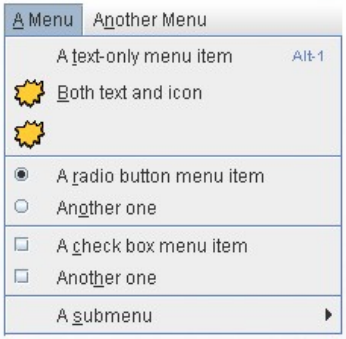
JCheckBox



JComboBox

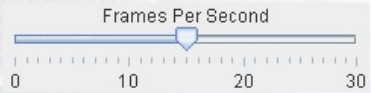


JList



JMenu

JRadioButton

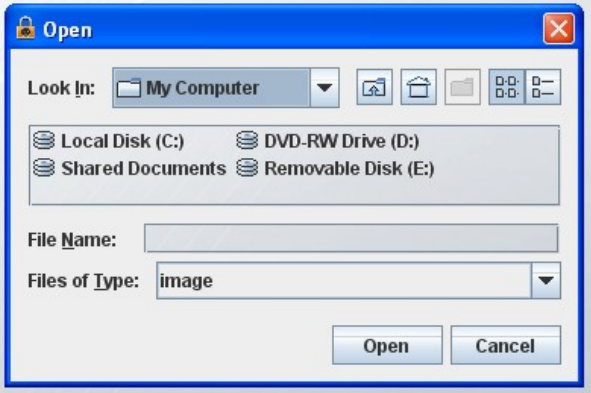


JSlider

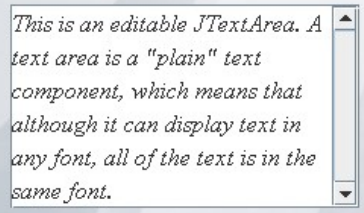
JSpinner

JTextField

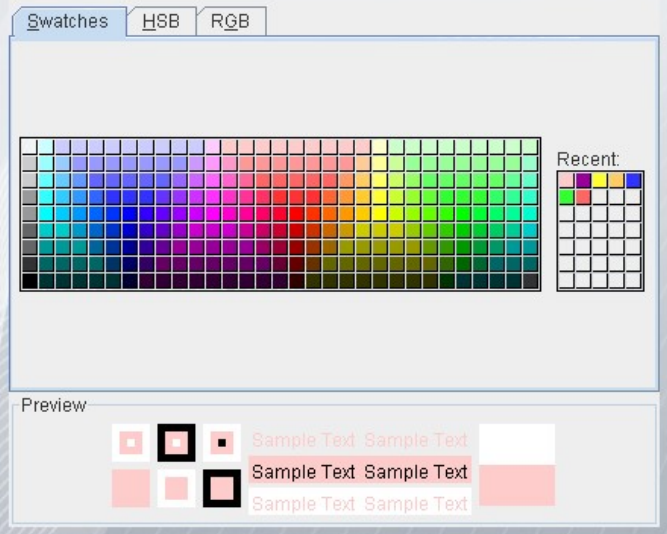
JPasswordField



JFileChooser



JTextArea



JColorChooser

Some swing components

- JFrame
- JPanel
- JLabel
- JTextField
- JButton
- JTextArea
- JScrollPane
- JComboBox
- JCheckBox

See the javadoc and investigate

Its methods and constructors

Demonstrating some swing components

```
public class ComponentsExample {
    public static void main(String[] args) {
        JFrame f = new JFrame("Components example");
        f.setSize(800, 600);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel p = new JPanel();
        f.add(p);

        JLabel l = new JLabel("Some text");
        p.add(l);

        JButton button = new JButton("Click here");
        p.add(button);
        button.setEnabled(true);

        JTextField field = new JTextField(20);
        p.add(field);
```

Demonstrating some swing components

```
        JTextArea te = new JTextArea(5, 10);
        //p.add(te);

        field.getText();
        te.getText();

        //te.setEnabled(false);
        te.setText("Text in the text area.\n Some text ...");

        JScrollPane sp = new JScrollPane(te);
        p.add(sp);

        Icon image = new ImageIcon("test.jpg");
        JOptionPane.showMessageDialog(null, "Text", "Title",
                                      JOptionPane.WARNING_MESSAGE, image);

        JComboBox c = new JComboBox();
        c.getSelectedItem();
        c.getSelectedIndex();
        f.setVisible(true);
    }
}
```