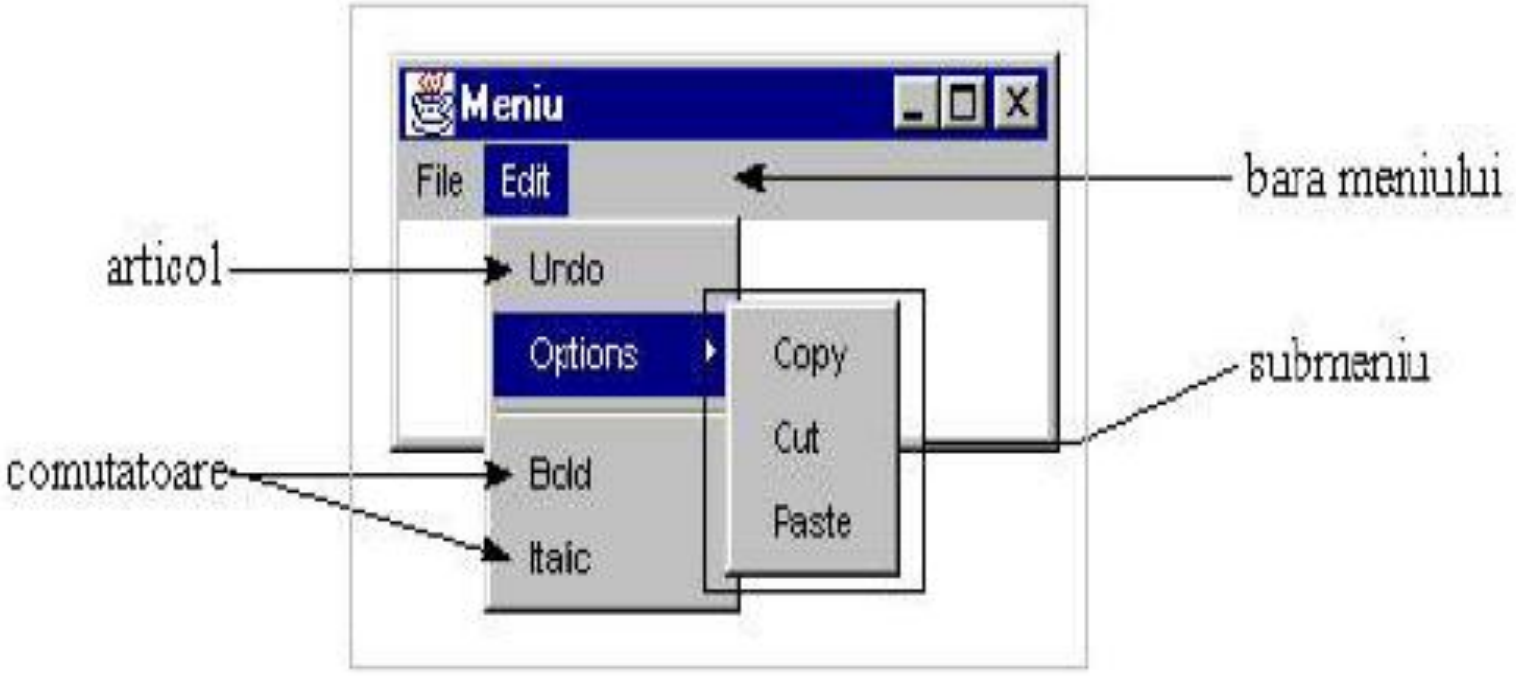
Interfața grafică cu utilizatorul - Meniuri

Programare Orientată pe Obiecte

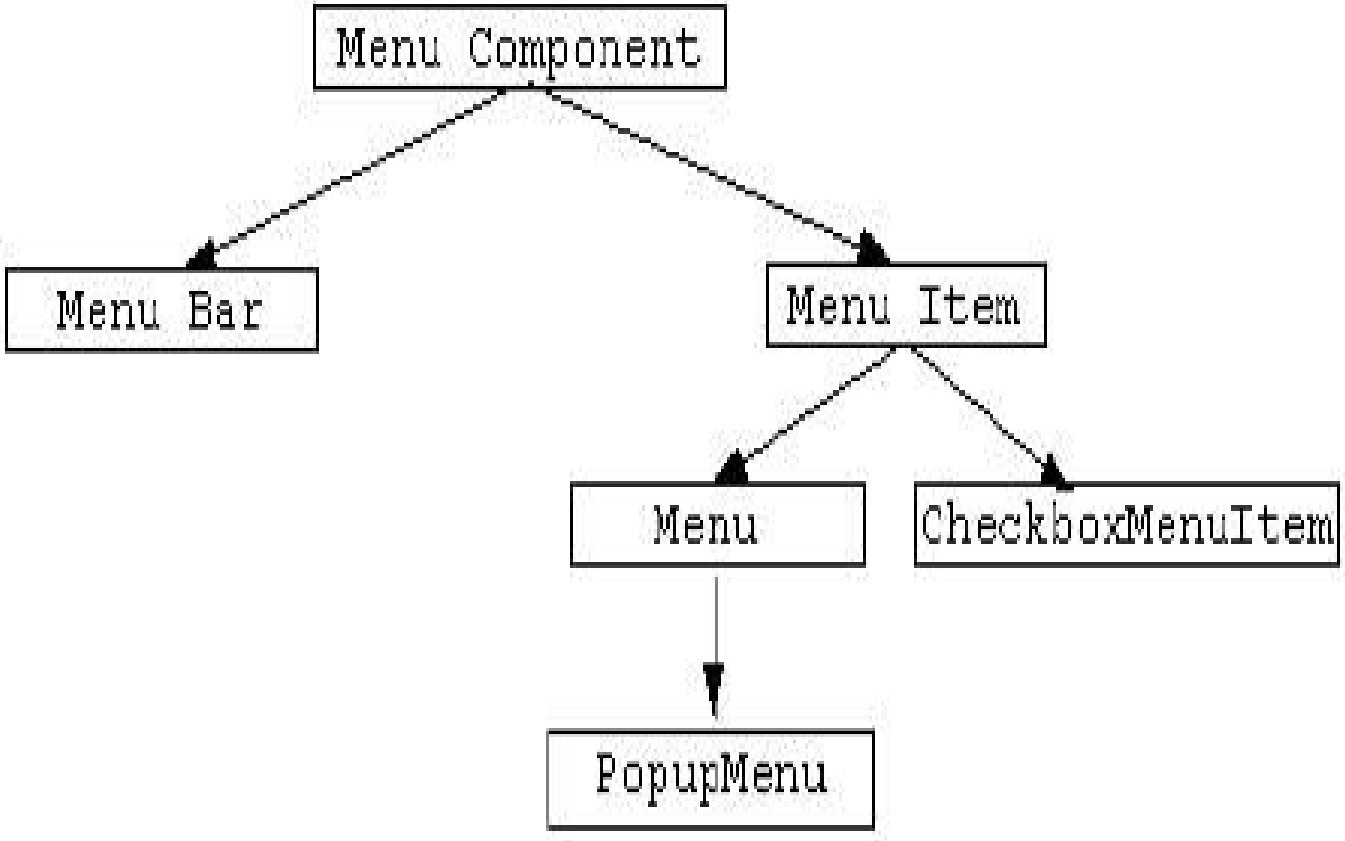
Folosirea meniurilor

- Componentele unui meniu sunt derivate din MenuComponent.
- Meniuri fixe (vizibile permanent)
- Meniuri de context (popup)



addMenuBar

Ierarhia claselor ce descriu meniuri



Exemplu

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

public class TestMenu {
   public static void main(String args[]) {
      Frame f = new Frame("Test Menu");

      MenuBar mb = new MenuBar();

      Menu fisier = new Menu("File");
      fisier.add(new MenuItem("Open"));
      fisier.add(new MenuItem("Close"));
      fisier.addSeparator();
      fisier.add(new MenuItem("Exit"));

      Menu optiuni = new Menu("Options");
      optiuni.add(new MenuItem("Copy"));
      optiuni.add(new MenuItem("Cut"));
      optiuni.add(new MenuItem("Paste"));

      Menu editare = new Menu("Edit");
      editare.add(new MenuItem("Undo"));
      editare.add(optiuni);

      editare.addSeparator();
      editare.add(new CheckboxMenuItem("Bold"));
      editare.add(new CheckboxMenuItem("Italic"));

      mb.add(fisier);
      mb.add(editare);

      f.setMenuBar(mb);
      f.setSize(200, 100);
      f.show();
   }
}
```

Tratarea evenimentelor generate de meniuri

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

public class TestMenuEvent extends Frame
     implements ActionListener, ItemListener {

  public TestMenuEvent(String titlu) {
    super(titlu);

    MenuBar mb = new MenuBar();
    Menu test = new Menu("Test");
    CheckboxMenuItem check = new CheckboxMenuItem("Check me")
        ;
    test.add(check);
    test.addSeparator();
    test.add(new MenuItem("Exit"));

    mb.add(test);
    setMenuBar(mb);

    Button btnExit = new Button("Exit");
    add(btnExit, BorderLayout.SOUTH);
    setSize(300, 200);
    show();
```

Tratarea evenimentelor generate de meniuri

```
        test.addActionListener(this);
        check.addItemListener(this);
        btnExit.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        // Valabila si pentru meniu si pentru buton
        String command = e.getActionCommand();
        if (command.equals("Exit"))
            System.exit(0);
    }

    public void itemStateChanged(ItemEvent e) {
        if (e.getStateChange() == ItemEvent.SELECTED)
            setTitle("Checked!");
        else
            setTitle("Not checked!");
    }

    public static void main(String args[]) {
        TestMenuEvent f = new TestMenuEvent("Tratare evenimente
            meniuri");
        f.show();
    }
}
```

Meniuri de context (popup)

```
PopupMenu popup = new PopupMenu("Options");
popup.add(new MenuItem("New"));
popup.add(new MenuItem("Edit"));
popup.addSeparator();
popup.add(new MenuItem("Exit"));
...
popup.show(Component origine, int x, int y)
fereastra.add(popup1);
...
fereastra.remove(popup1);
fereastra.add(popup2);
```

isPopupTrigger() :

- dacă acest eveniment de mouse este evenimentul care poate afişa un meniu popup
- trebuie testat în ambele metode mousePressed şi mouseReleased pentru că depinde de platformă

Folosirea unui meniu de context (popup)

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

class Fereastra extends Frame implements ActionListener{
  // Definim meniul popup al ferestrei
  private PopupMenu popup;
  // Pozitia meniului va fi relativa la fereastra
  private Component origin;
  public Fereastra(String titlu) {
    super(titlu);
    origin = this;

    this.addWindowListener(new WindowAdapter() {
      public void windowClosing(WindowEvent e) {
        System.exit(0);
      }
    });

    this.addMouseListener(new MouseAdapter() {
      public void mousePressed(MouseEvent e) {
        if (e.isPopupTrigger())
          popup.show(origin, e.getX(), e.getY());
      }
      public void mouseReleased(MouseEvent e) {
        if (e.isPopupTrigger())
          popup.show(origin, e.getX(), e.getY());
      }
    });
    setSize(300, 300);
```

Folosirea unui meniu de context (popup) (2)

```
    // Cream meniul popup
    popup = new PopupMenu("Options");
    popup.add(new MenuItem("New"));
    popup.add(new MenuItem("Edit"));
    popup.addSeparator();
    popup.add(new MenuItem("Exit"));
    add(popup); //atasam meniul popup ferestrei
    popup.addActionListener(this);
}

public void actionPerformed(ActionEvent e) {
    String command = e.getActionCommand();
    if (command.equals("Exit"))
        System.exit(0);
    }
}

public class TestPopupMenu {
    public static void main(String args[]) {
        Fereastra f = new Fereastra("PopupMenu");
        f.show();
    }
}
```

Acceleratori

Clasa MenuShortcut

Ctrl + Tasta sau Ctrl + Shift + Tasta

```
// Ctrl+O
new MenuItem("Open",new MenuShortcut(
    KeyEvent.VK_O));

// Ctrl+P
new MenuItem("Print",new MenuShortcut('p'));

// Ctrl+Shift+P
new MenuItem("Preview",new MenuShortcut('p'),
    true);
```