GUI Programming

- Reading: Savitch, Chapter 12

Components, Containers, and Layout Managers in AWT(cont)

Numerical Input and Output

- The following example shows the way to conduct numerical input and output through GUI.

Example

```
//Adder.java
import java.awt.*;
import java.awt.event.*;
/* numerical input and output */
public class Adder extends Frame implements
    ActionListener {
    public static final int WIDTH = 300;
    public static final int HEIGHT = 300;
    public static final int X = 80;
    public static final int Y = 50;

    private TextField inputOutputField;
    private double sum = 0;
```

```
public static void main(String[] args) {
    Adder guiAdder = new Adder();
    guiAdder.setVisible(true);
}

public Adder() {
    setTitle("Adding Machine");
    addWindowListener(new WindowDestroyer());
    setSize(WIDTH, HEIGHT);
    setLocation(X,Y);

    setLayout(new BorderLayout());
```

```
Panel buttonPanel = new Panel();
buttonPanel.setBackground(Color.gray);
buttonPanel.setLayout(new FlowLayout());
Button addButton = new Button("Add In");
addButton.addActionListener(this);
buttonPanel.add(addButton);
Button resetButton = new Button("Reset");
resetButton.addActionListener(this);
buttonPanel.add(resetButton);
add(buttonPanel, "South");
```

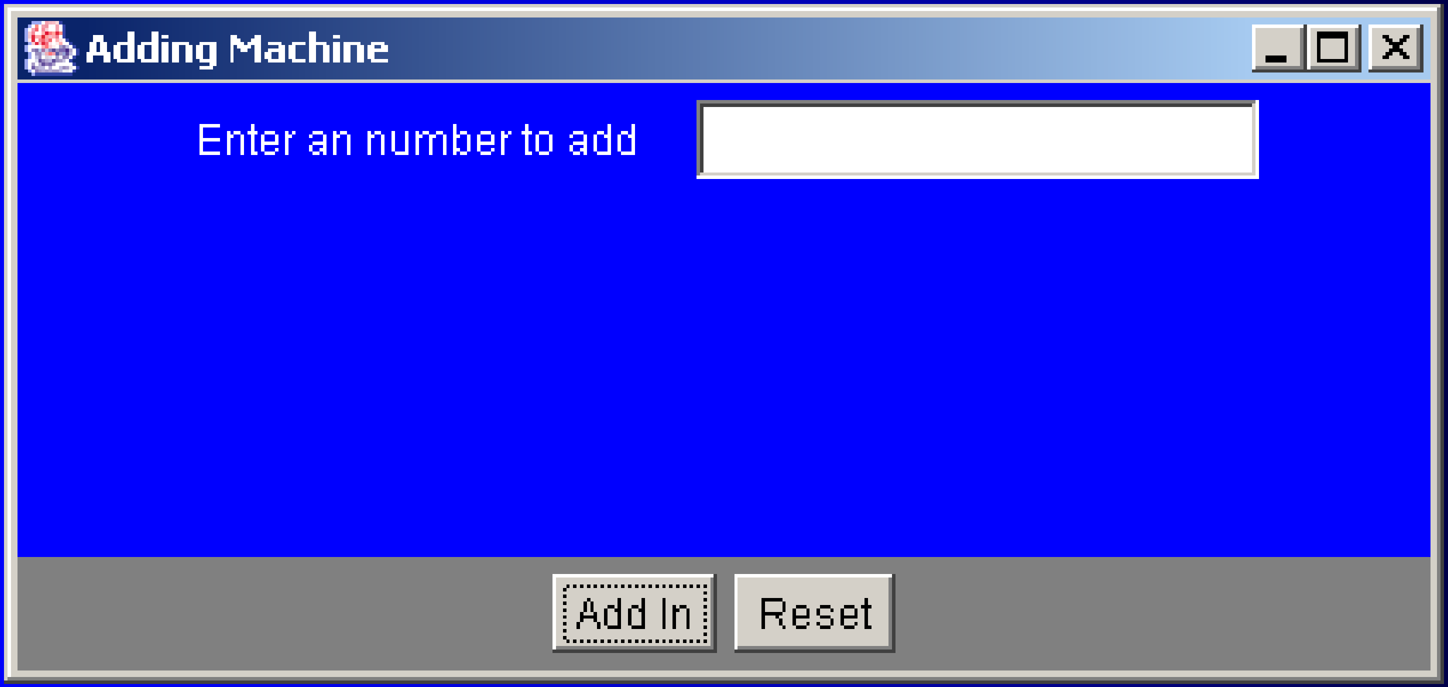
```
Panel textPanel = new Panel();
textPanel.setBackground(Color.blue);
inputOutputField = new TextField(20);
inputOutputField.setBackground(Color.white);
Label prompt = new Label("Enter a number to add");
prompt.setForeground(Color.white);

textPanel.add(prompt, "Left");
textPanel.add(inputOutputField);
add(textPanel, "Center");
}
```

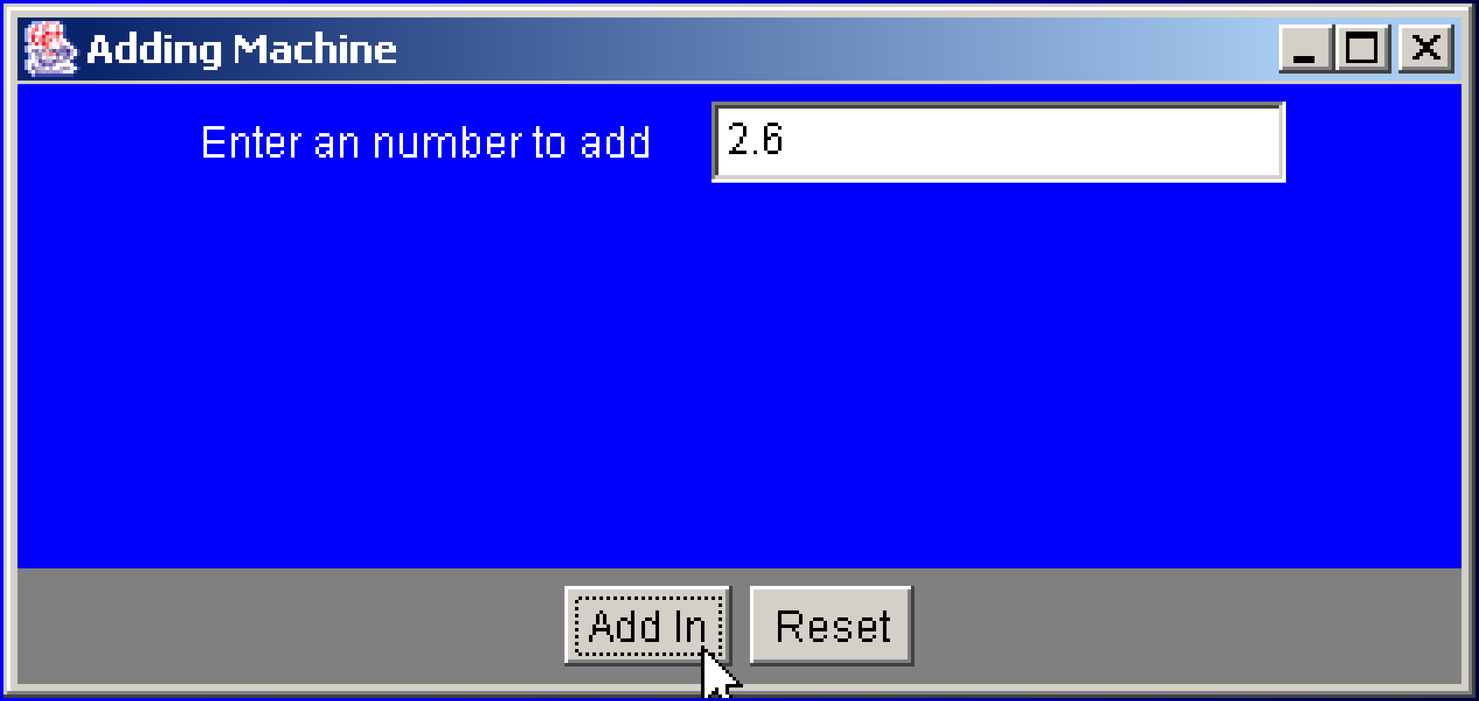
```
public void actionPerformed(ActionEvent e) {
    if (e.getActionCommand().equals("Add In")) {
        sum = sum +
            stringToDouble(inputOutputField.getText());
        inputOutputField.setText(Double.toString(sum));
    }
    else if (e.getActionCommand().equals("Reset")) {
        sum = 0;
        inputOutputField.setText("0.0");
    }
    else
        inputOutputField.setText("Error in adder code.");
    repaint();
  }
```

```
private static double stringToDouble(String stringObject) {
  return Double.valueOf(stringObject.trim()).doubleValue();
  }
}
```

Program execution

Java Adder

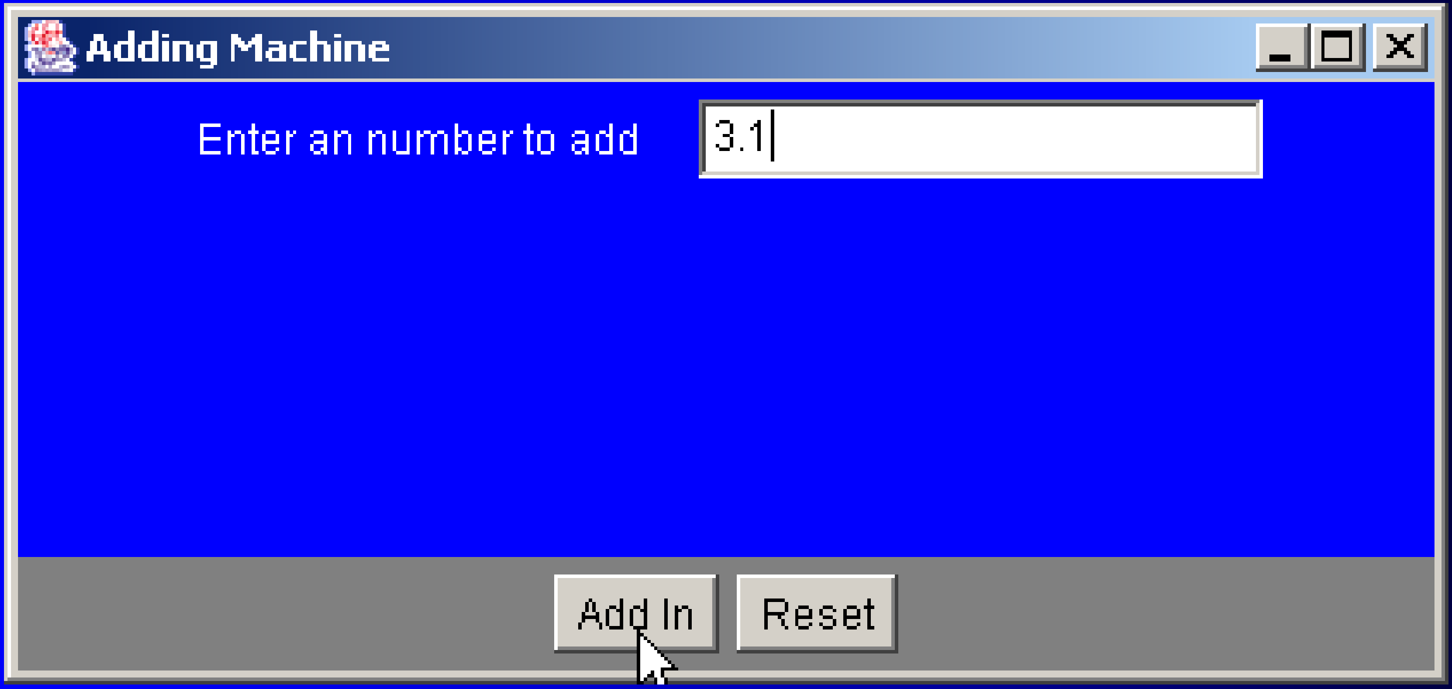




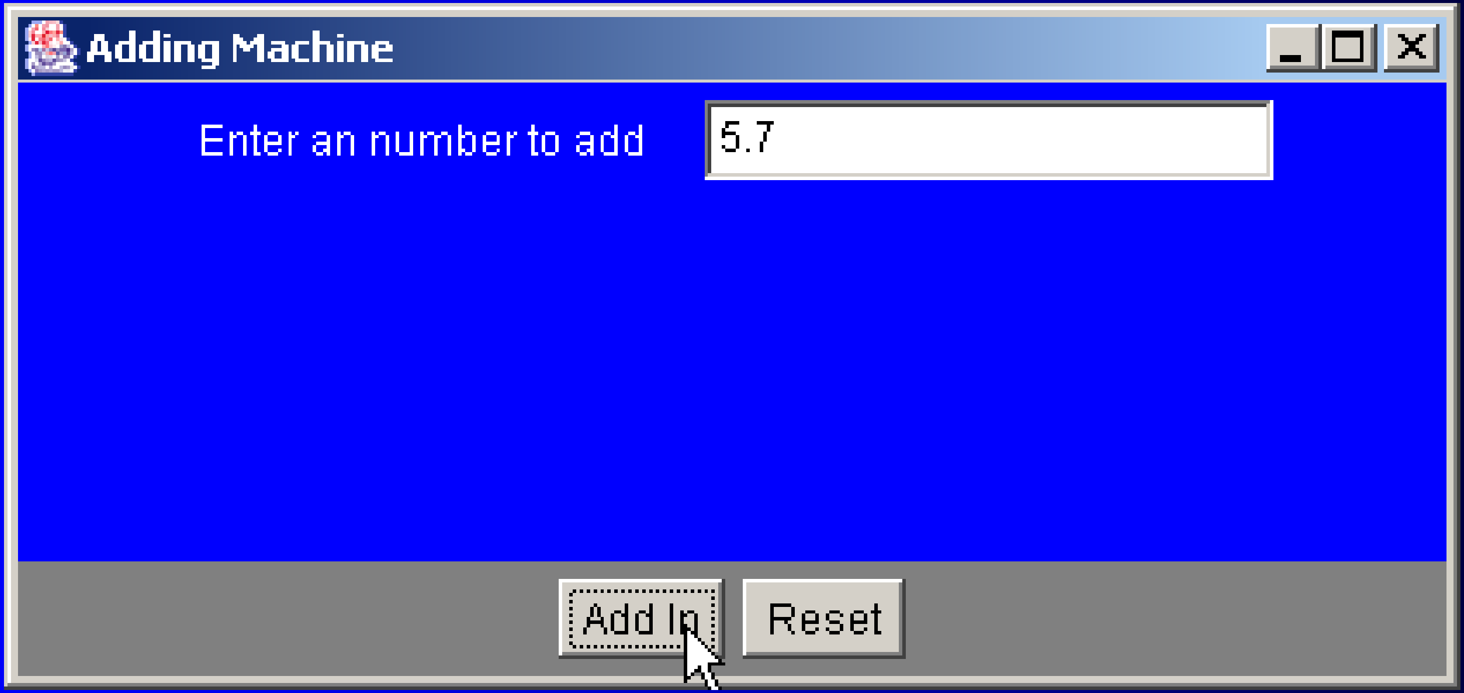
Adding Machine

Enter an number to add

2.6



Adding Machine
Enter an number to add
3.1
Add In
Reset



Adding Machine

Enter an number to add 5.7

Some notes

- The example shows a way to take numerical input from a text field.

```
private static double stringToDouble(String stringObject) {
        return
    Double.valueOf(stringObject.trim()).doubleValue();
}

stringToDouble(inputOutputField.getText());
```

- And a way to display a numerical value on a text field.

```
inputOutputField.setText(Double.toString(sum));
```

- The Label class is defined in API. It is an object which contains a line of text that can be added into a container class.
- For example:

```
Label prompt = new Label("Enter an number to add");
prompt.setForeground(Color.white);
textPanel.add(prompt, "Left");
```

MenuBar, Menu and MenuItem

- The following example presents the way to build up a menu system inside a frame.

Example

```
//MenuDemo.java
import java.awt.*;
import java.awt.event.*;
/* save and retrieve memos using a menu. (by W.
    Savitch)*/
public class MenuDemo extends Frame implements
    ActionListener {
    public static final int WIDTH = 500;
    public static final int HEIGHT = 300;
    public static final int X = 100;
    public static final int Y = 30;
    private TextArea theText;
    private String memo1 = "No Memo 1.";
    private String memo2 = "No Memo 2.";
```

```
public MenuDemo() {
        setTitle("Memo Saver");
        setSize(WIDTH, HEIGHT);
        setLocation(X, Y);
        setBackground(Color.blue);
        addWindowListener(new WindowDestroyer());

    Menu comm = new Menu("System Commands");
        MenuItem m;
        m = new MenuItem("Clear");
        m.addActionListener(this);
        comm.add(m);
    m = new MenuItem("Exit");
    m.addActionListener(this);
    comm.add(m);

Menu saveMemo = new Menu("Save Memos");
    m = new MenuItem("Save Memo 1");
    m.addActionListener(this);
    saveMemo.add(m);

    m = new MenuItem("Save Memo 2");
    m.addActionListener(this);
    saveMemo.add(m);
```

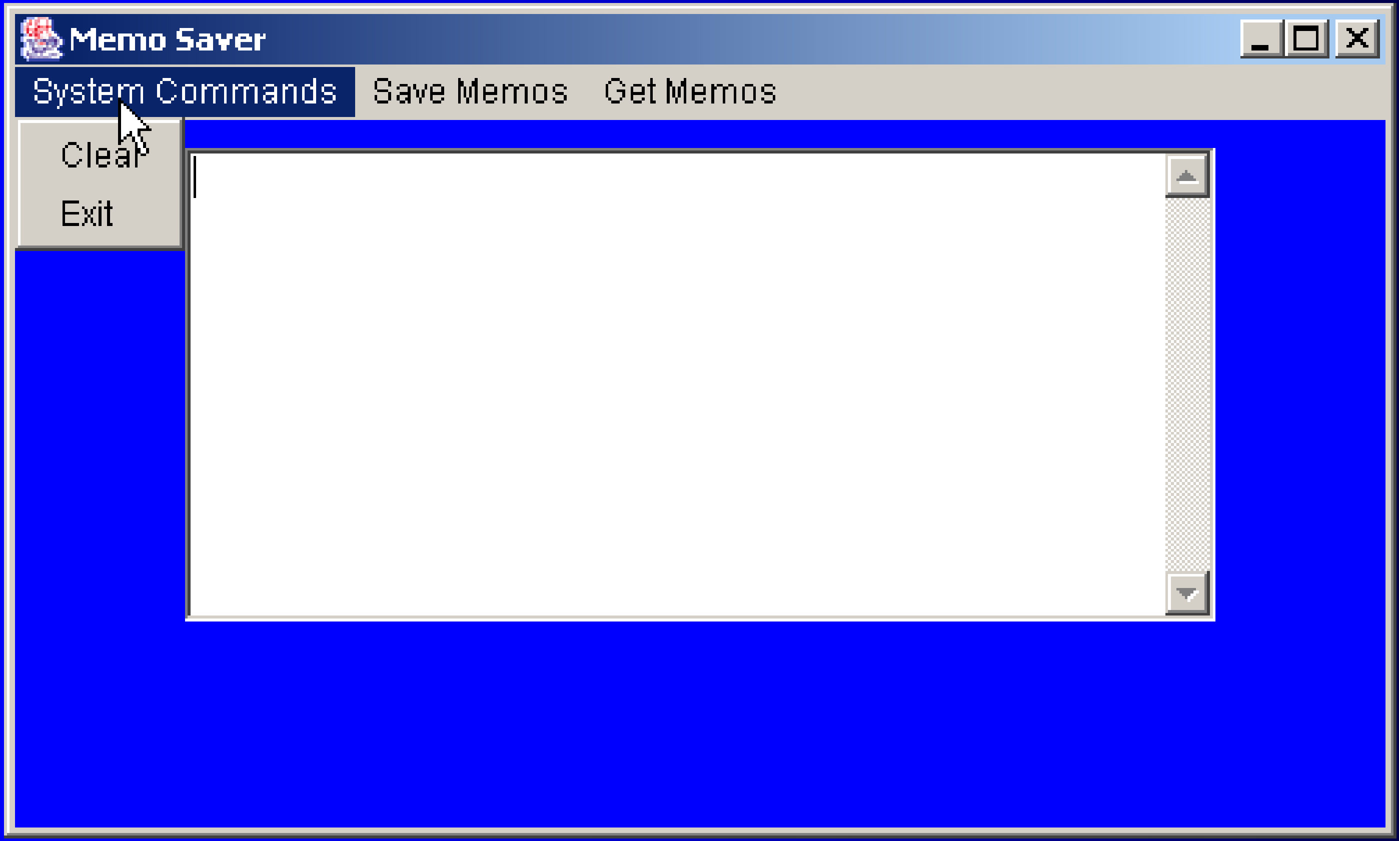
```
Menu getMemo = new Menu("Get Memos");
    m = new MenuItem("Get Memo 1");
    m.addActionListener(this);
    getMemo.add(m);
    m = new MenuItem("Get Memo 2");
    m.addActionListener(this);
    getMemo.add(m);

    MenuBar mBar = new MenuBar();
    mBar.add(comm);
    mBar.add(saveMemo);
    mBar.add(getMemo);
    setMenuBar(mBar);
```

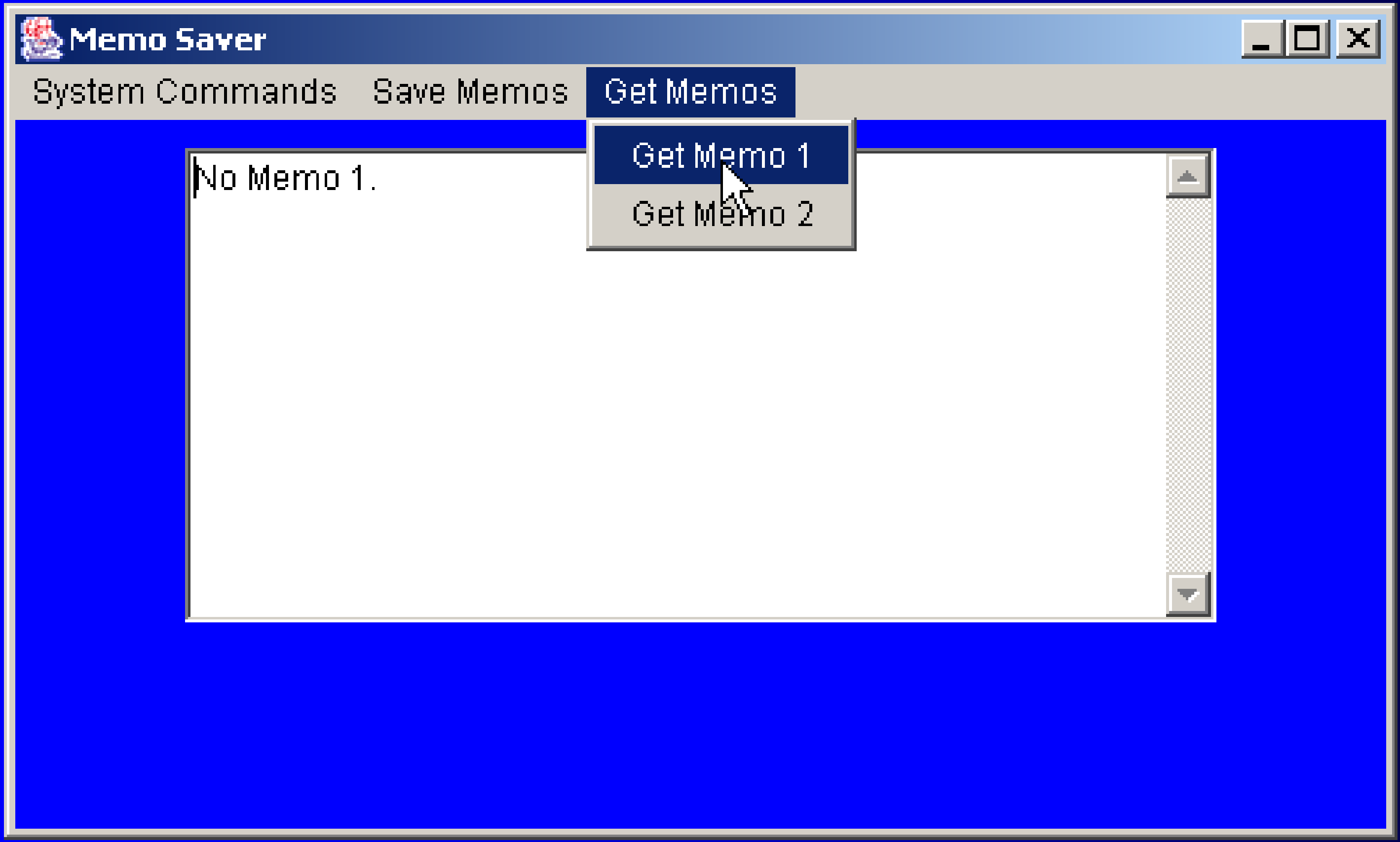
```
         Panel textPanel = new Panel();
        textPanel.setBackground(Color.blue);
        theText = new TextArea(10, 50);
        theText.setBackground(Color.white);
        textPanel.add(theText);
        setLayout(new FlowLayout());
        add(textPanel);
      }
```

```
public void actionPerformed(ActionEvent e) {
    String actionCommand = e.getActionCommand();
    if (actionCommand.equals("Save Memo 1"))
      memo1 = theText.getText();
    else if (actionCommand.equals("Save Memo 2"))
      memo2 = theText.getText();
    else if (actionCommand.equals("Clear"))
      theText.setText("");
    else if (actionCommand.equals("Get Memo 1"))
      theText.setText(memo1);
    else if (actionCommand.equals("Get Memo 2"))
      theText.setText(memo2);
else if (actionCommand.equals("Exit"))
            System.exit(0);
        else
            theText.setText("Error in memo interface.");

       repaint();
   }

    public static void main(String[] args) {
        MenuDemo menuGUI = new MenuDemo();
        menuGUI.setVisible(true);
    }
}
```

Program execution

Java MenuDemo

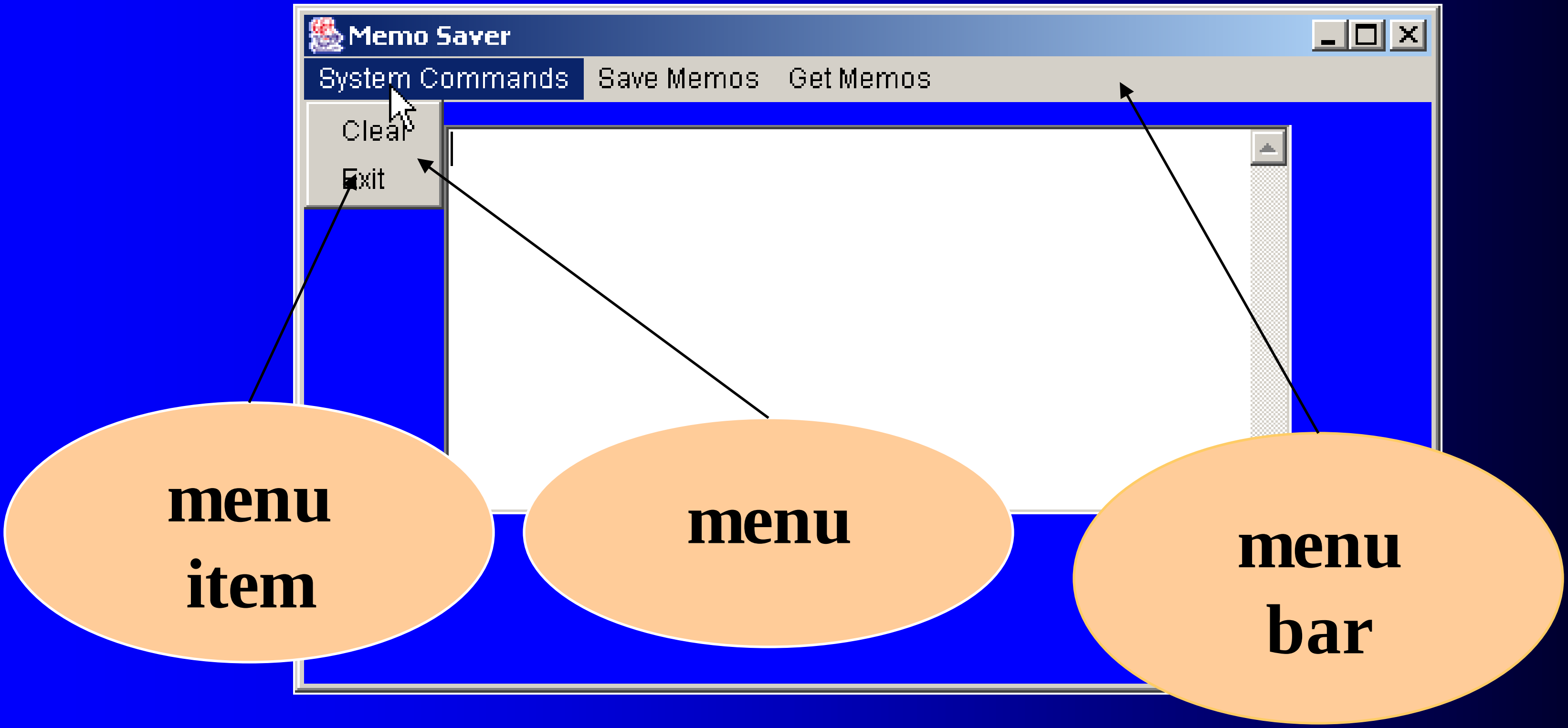



Memo Saver
System Commands
Save Memos
Get Memos
Get Memo 1
Get Memo 2

No Memo 1.

Some note

- MenuBar, Menu and MenuItem (built in API) are the three classes we normally use to establish our menu system.



- We add menu items onto a menu; add menus onto a menu bar; and add the menu bar to the frame. We then link an action with each menu item.