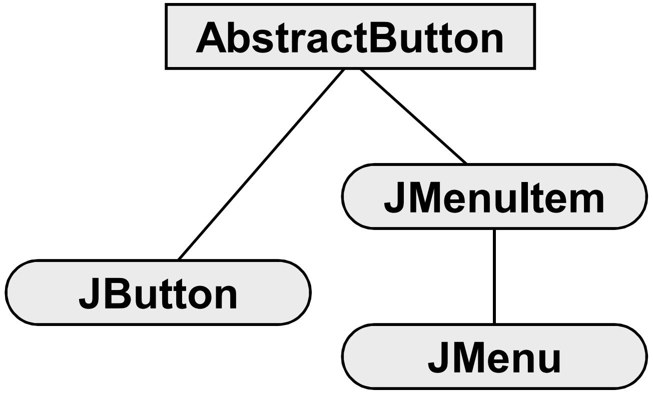
Chapter 15

More Swing

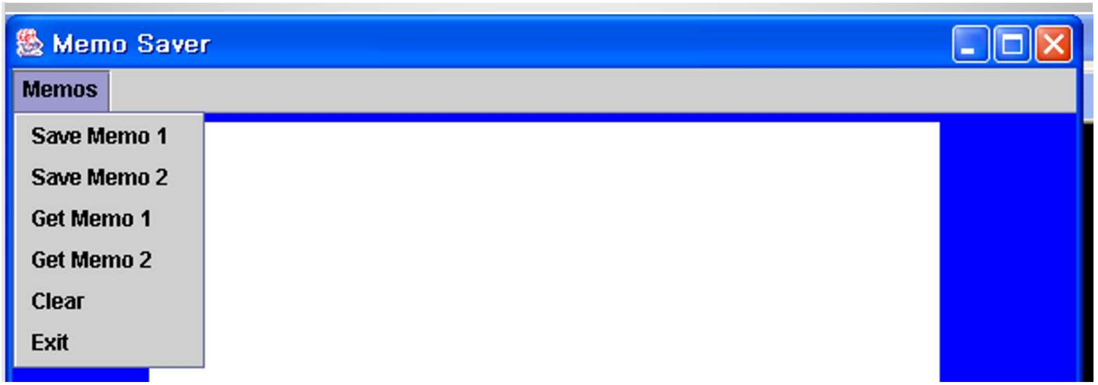
- 15.1 Menus
- 15.2 Making GUIs Pretty (and More Functional)
- 15.3 More Layout Managers
- 15.4 Inner Classes
- 15.5 More on Events and Listeners
- 15.6 Another Look at the Swing Class Hierarchy

15.1 Menus

- Three Swing classes used to put a menu in a program:
  - »
  - »
  - »



- Menu items behave in the same way as buttons



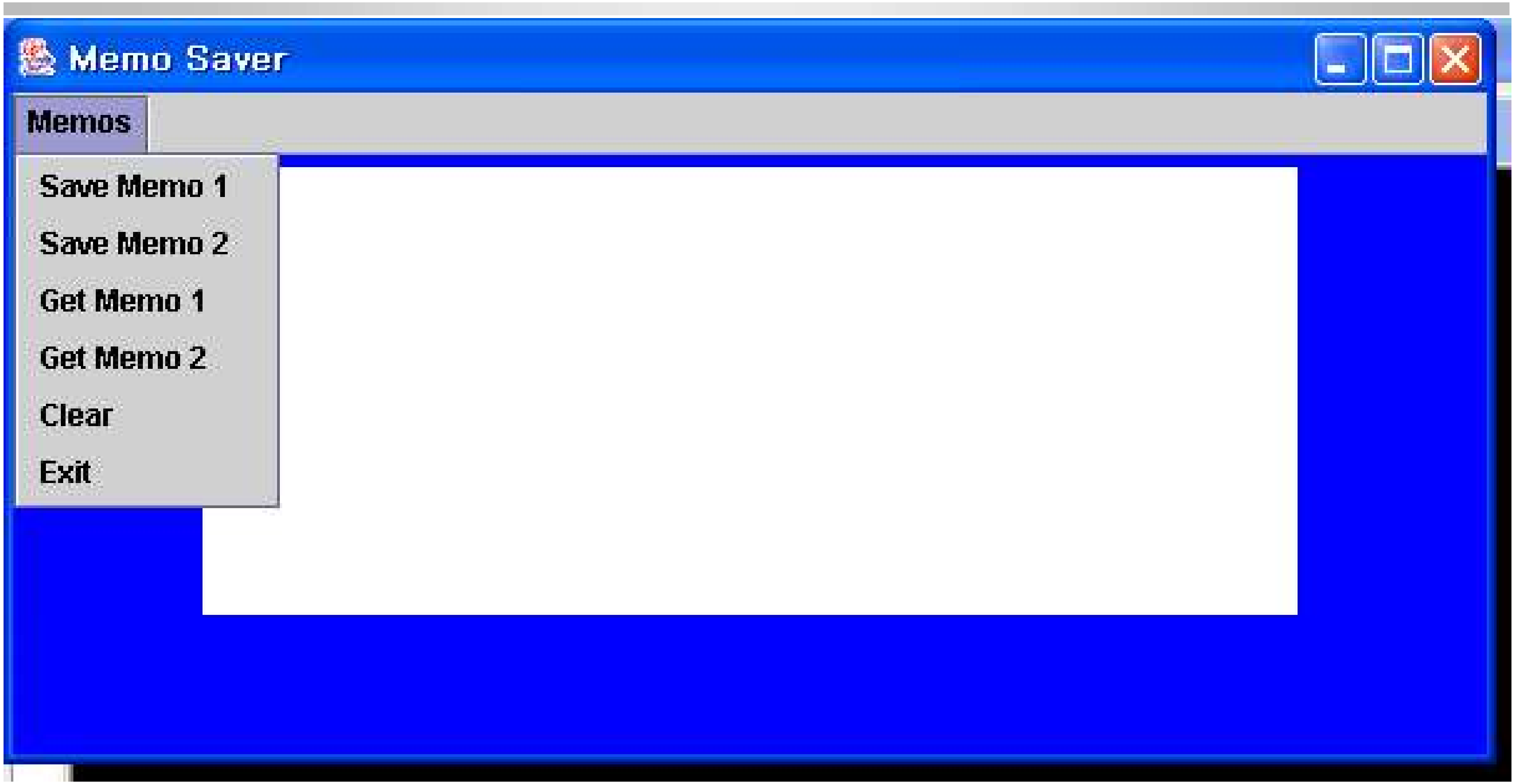
Listing 15.1 A GUI with a Menu - MemoGUI.java

```
// Listing 15.1 A GUI with a Menu


import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class MemoGUI extends JFrame implements ActionListener
{
    public static final int WIDTH = 600;
    public static final int HEIGHT = 300;
    public static final int LINES = 10;
    public static final int CHAR_PER_LINE = 40;

    private JTextArea theText;
    private String memo1 = "No Memo 1.";
    private String memo2 = "No Memo 2.";
```

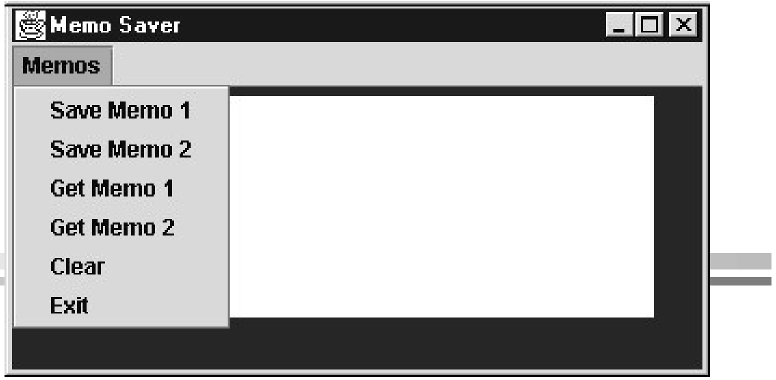
```
public MemoGUI( )
   {
       setSize(WIDTH, HEIGHT);
       addWindowListener(new WindowDestroyer( ));
     // THe Class WindowDestroyer is defined and discussed in Chapter 12

       setTitle("Memo Saver");
       Container contentPane = getContentPane( );
       contentPane.setLayout(new BorderLayout( ));

                         //
       JMenu memoMenu = new JMenu("Memos");
       JMenuItem m;

       m = new JMenuItem("Save Memo 1");
       m.addActionListener(this);
       memoMenu.add(m);
                         //

       m = new JMenuItem("Save Memo 2");
       m.addActionListener(this);
       memoMenu.add(m);

       m = new JMenuItem("Get Memo 1");
       m.addActionListener(this);
       memoMenu.add(m);
```

```
    m = new JMenuItem("Get Memo 2");
    m.addActionListener(this);
    memoMenu.add(m);

    m = new JMenuItem("Clear");
    m.addActionListener(this);
    memoMenu.add(m);


    m = new JMenuItem("Exit");
    m.addActionListener(this);
    memoMenu.add(m);

                    //
    JMenuBar mBar = new JMenuBar( );
    mBar.add(memoMenu);
    setJMenuBar(mBar);

    JPanel textPanel = new JPanel( );
    textPanel.setBackground(Color.BLUE);
    theText = new JTextArea(LINES, CHAR_PER_LINE);
    theText.setBackground(Color.WHITE);
    textPanel.add(theText);
    contentPane.add(textPanel, BorderLayout.CENTER);
}
```

```
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");
  }

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


Memo Saver
Memos
Save Memo 1
Save Memo 2
Get Memo 1
Get Memo 2
Clear
Exit

A GUI with a Menu



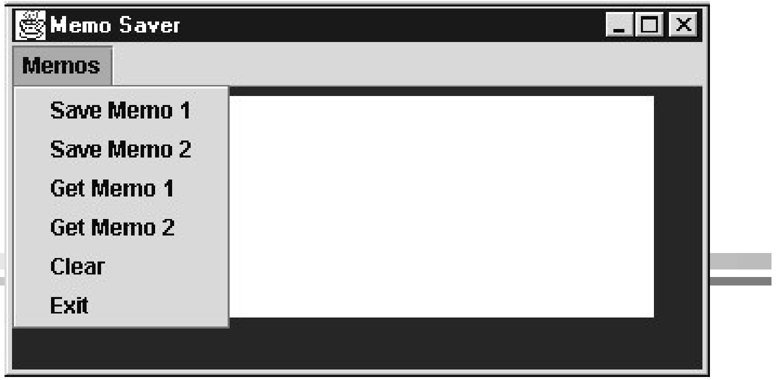
```
JMenu memoMenu = new JMenu("Memos");
JMenuItem m;
m = new JMenuItem("Save Memo 1");
m.addActionListener(this);
memoMenu.add(m);
m = new JMenuItem("Save Memo 2");
m.addActionListener(this);
memoMenu.add(m);
. . .
JMenuBar mBar = new JMenuBar();
mBar.add(memoMenu);
setJMenuBar(mBar);
```

Create a menu

Create a menu item

A menu item uses an action listener the same way a button does.

A GUI with a Menu



```
JMenu memoMenu = new JMenu("Memos");
JMenuItem m;
m = new JMenuItem("Save Memo 1");
m.addActionListener(this);
memoMenu.add(m);
m = new JMenuItem("Save Memo 2");
m.addActionListener(this);
memoMenu.add(m);
. . .
JMenuBar mBar = new JMenuBar();
mBar.add(memoMenu);
setJMenuBar(mBar);
```

Each menu item is added to the menu.

The menu is added to the menu bar.

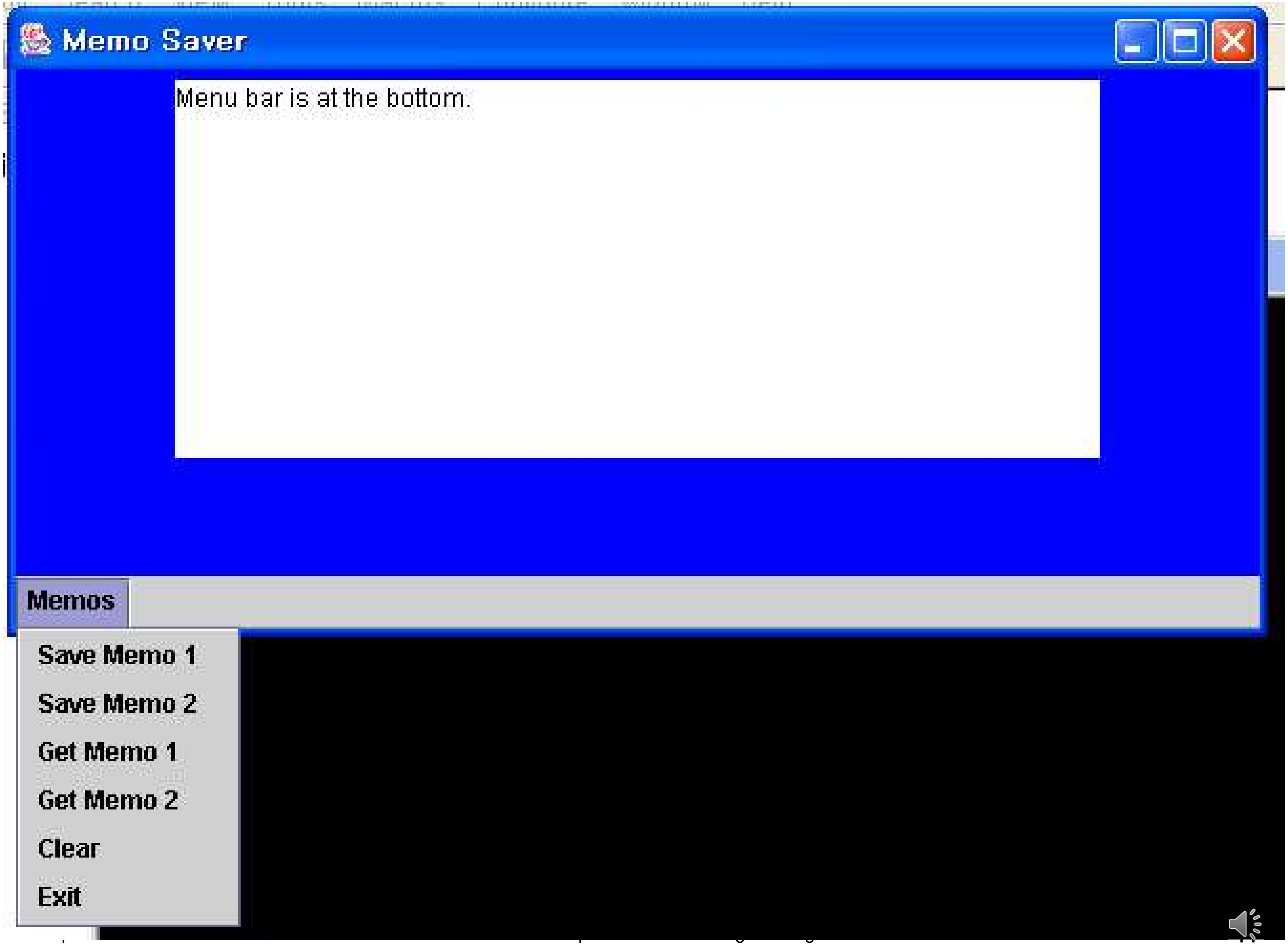
One way to add a menu bar to a `JFrame`

add a menu bar to a `JFrame`

```
JMenuBar mBar = new JMenuBar();
mBar.add(memoMenu);
setJMenuBar(mBar);
```

`MenuAdd.java`

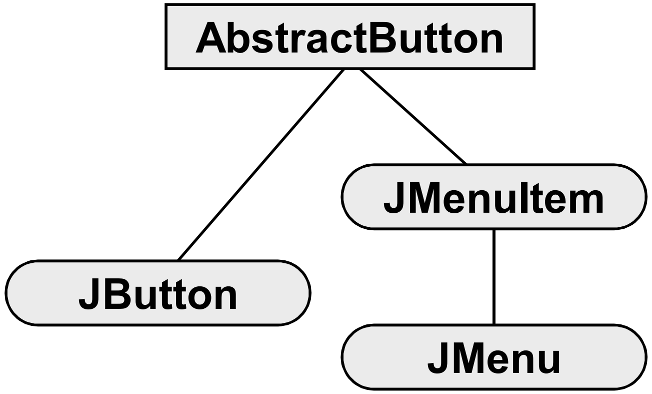
```
JMenuBar mBar = new JMenuBar();
mBar.add(memoMenu);
```



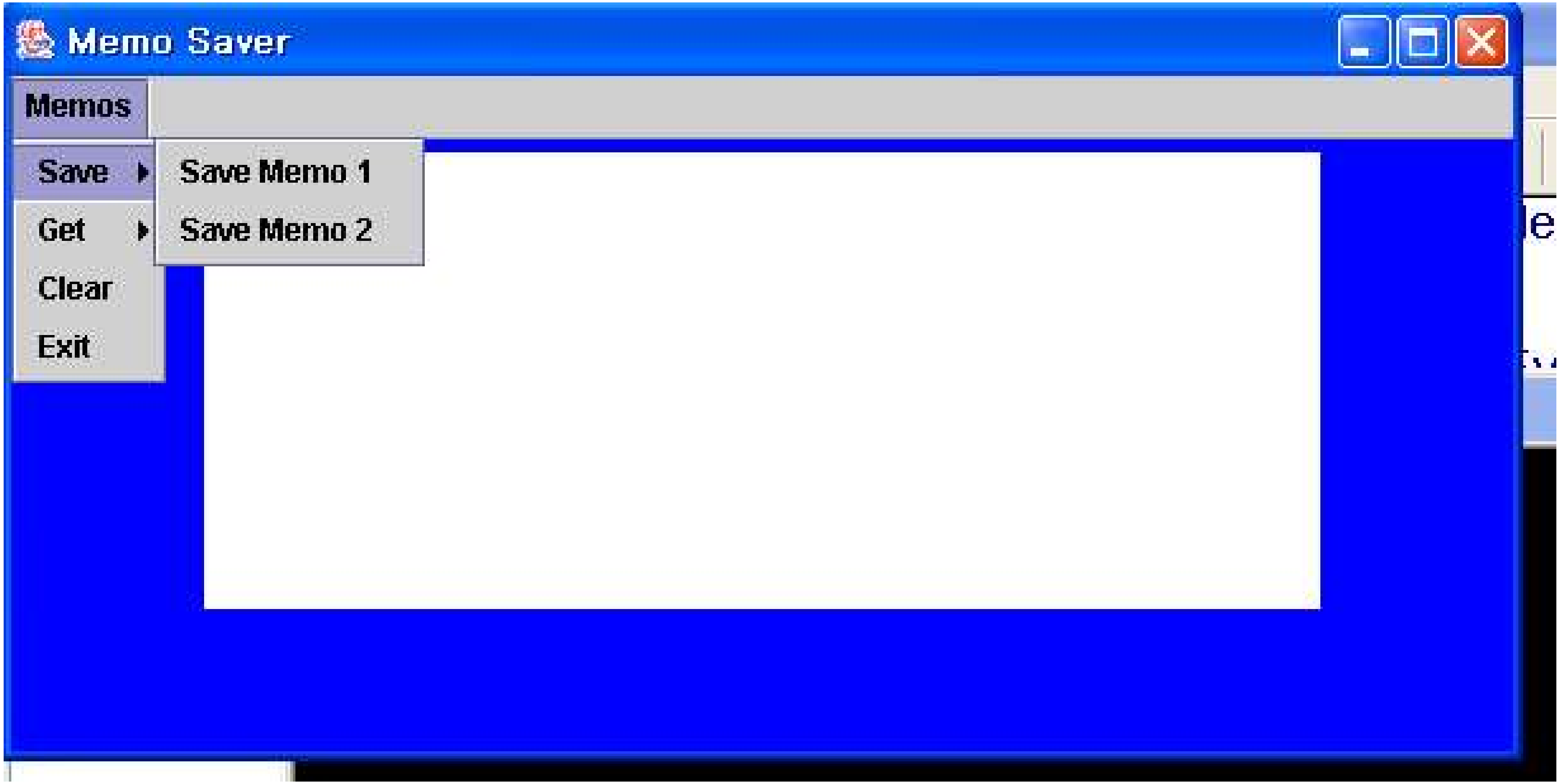
Menu bar is at the bottom.

Nested Menus

- `JMenu` is a descendant of `JMenuItem`
- Every `JMenu` object is also a `JMenuItem`
- A `JMenu` can be a menu item in another menu
- This allows nested menus
- Clicking on a nested menu shows the items in the nested menu and allows them to be selected.



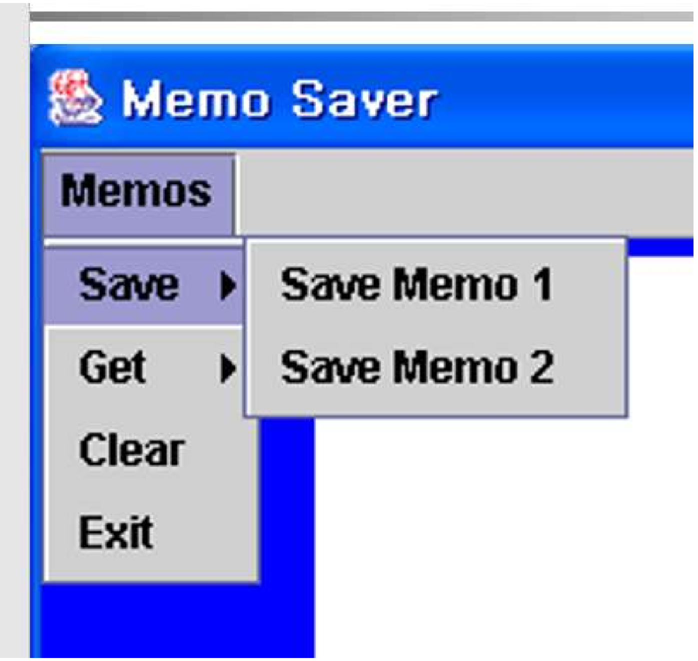
Nested Menus - NestedMenus.java



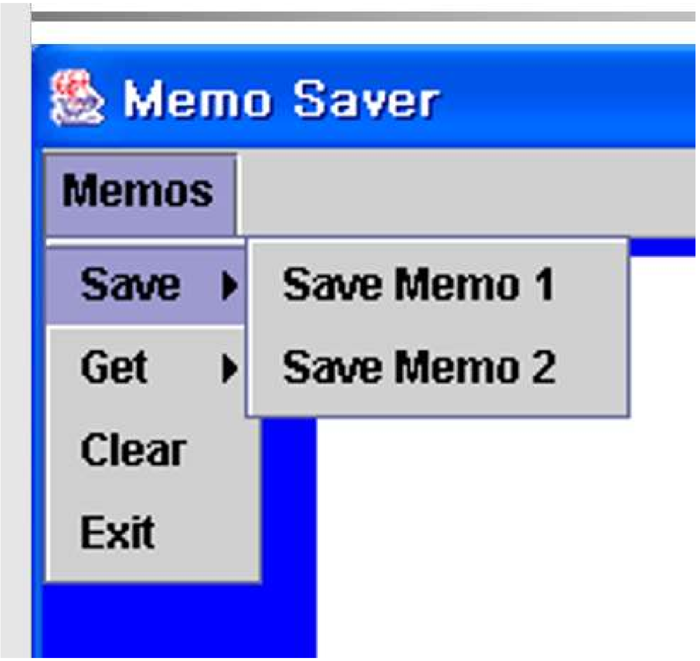
Nested Menus - NestedMenus.java

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

public class NestedMenus extends JFrame implements ActionListener
{
    public static final int WIDTH = 600;
    public static final int HEIGHT = 300;
    public static final int LINES = 10;
    public static final int CHAR_PER_LINE = 40;

    private JTextArea theText;
    private String memo1 = "No Memo 1.";
    private String memo2 = "No Memo 2.";
```

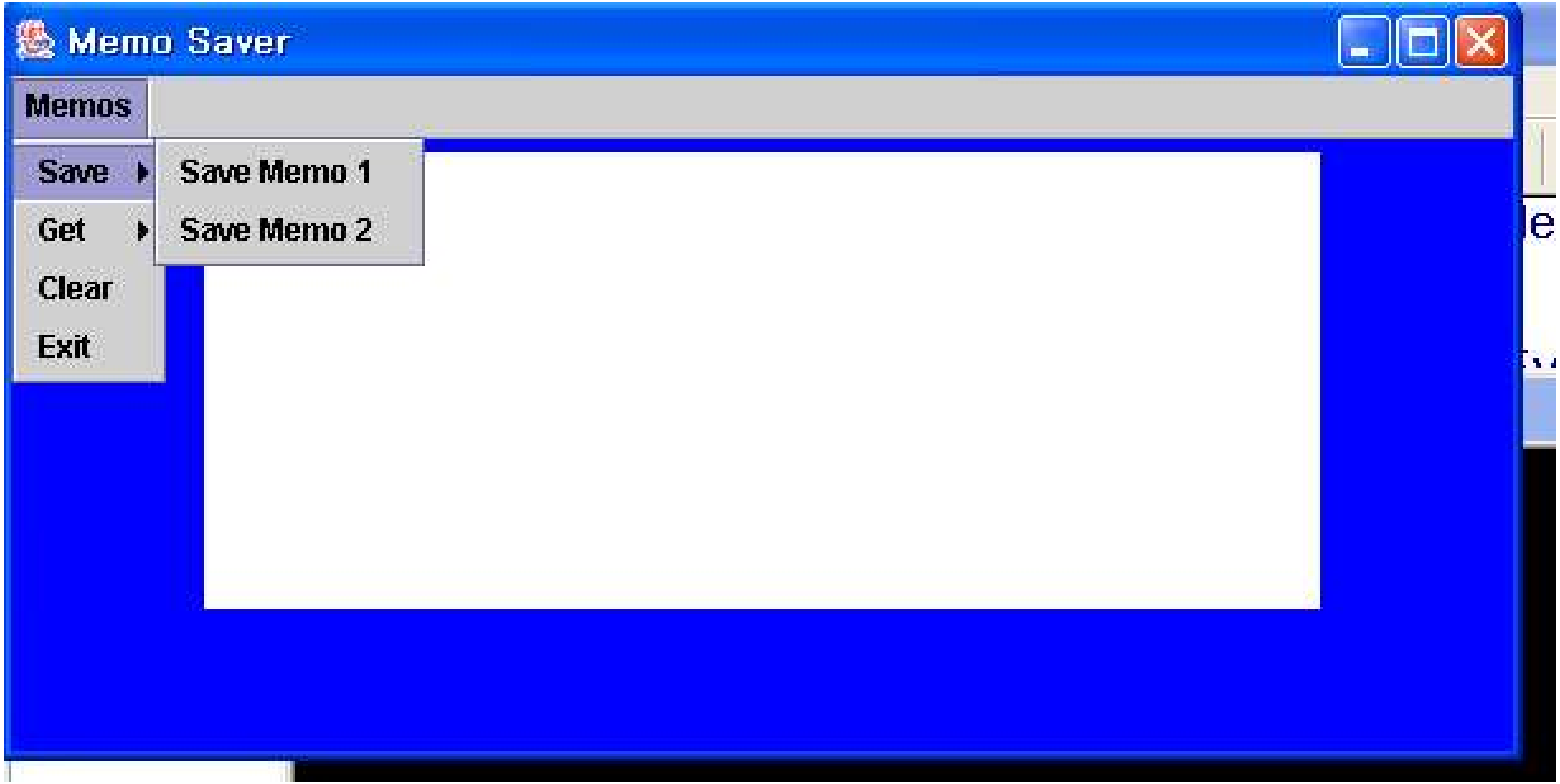
```
public NestedMenus()
  {
     setSize(WIDTH, HEIGHT);
     addWindowListener(new WindowDestroyer());
     setTitle("Memo Saver");
     Container contentPane = getContentPane();
     contentPane.setLayout(new BorderLayout());

     JMenu memoMenu = new JMenu("Memos");
     JMenuItem m;

     JMenu saveMenu = new JMenu("Save");

     m = new JMenuItem("Save Memo 1");
     m.addActionListener(this);
     saveMenu.add(m);

     m = new JMenuItem("Save Memo 2");
     m.addActionListener(this);
     saveMenu.add(m);

     memoMenu.add(saveMenu);

     JMenu getMenu = new JMenu("Get");

     m = new JMenuItem("Get Memo 1");
     m.addActionListener(this);
     getMenu.add(m);
```



```
m = new JMenuItem("Get Memo 2");
      m.addActionListener(this);
      getMenu.add(m);

      memoMenu.add(getMenu);

      m = new JMenuItem("Clear");
      m.addActionListener(this);
      memoMenu.add(m);

      m = new JMenuItem("Exit");
      m.addActionListener(this);
      memoMenu.add(m);

      JMenuBar mBar = new JMenuBar();
      mBar.add(memoMenu);
      setJMenuBar(mBar);

      JPanel textPanel = new JPanel();
      textPanel.setBackground(Color.BLUE);
      theText = new JTextArea(LINES, CHAR_PER_LINE);
      theText.setBackground(Color.WHITE);
      textPanel.add(theText);
      contentPane.add(textPanel, BorderLayout.CENTER);
  }
```



```
 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");
   }

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



Memo Saver
Memos
Save
Get
Clear
Exit
Save Memo 1
Save Memo 2

끝