Programare avansata pe obiecte – laborator 12

Alina Puscasu alina.puscasu@endava.com

Swing Graphical User Interface

Java provides a set of components 'Swing components' which allow the creation of a User Interface that works across different platforms (e.g. Mac / Windows).

Creating a sample project:

1. Create a new Java Project containing a package called gui.
2. Right click on the gui folder in the project window and choose: New -> Swing UI Designer -> GUI Form
3. Name the form 'HelloWorld', and configure it as described in the following steps.
4. The form will be displayed and will include a panel. In the component tree (select the JPanel and enter the value 'rootPanel' into the field name property box:
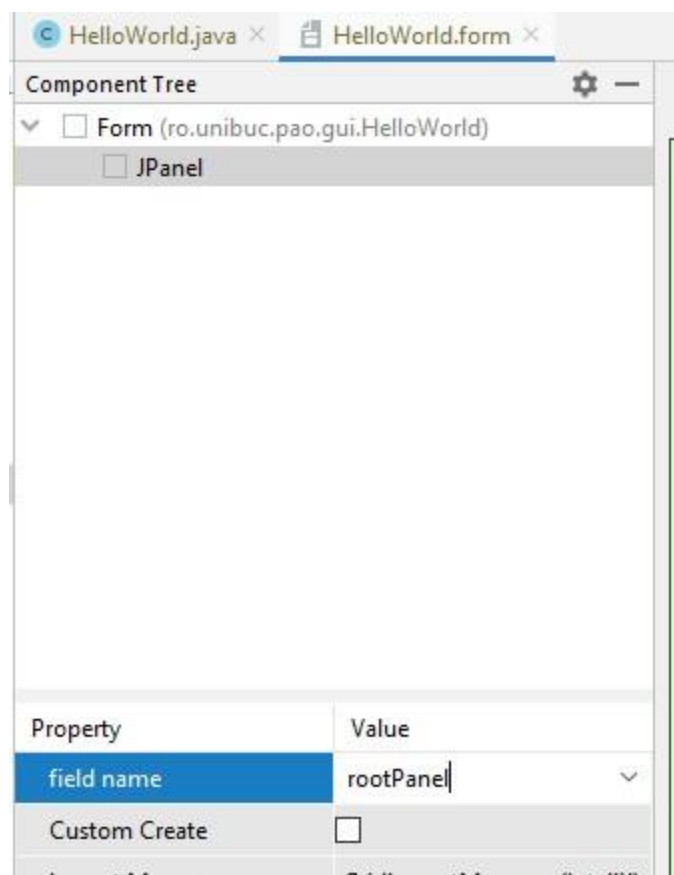
5. Navigate to the HelloWorld.java file and ensure that you can now see the jPanel as a variable in the HelloWorld class as follows:

```java
package ro.unibuc.pao.gui;

import javax.swing.*;

public class HelloWorld {
    private JPanel rootPanel;
}
```

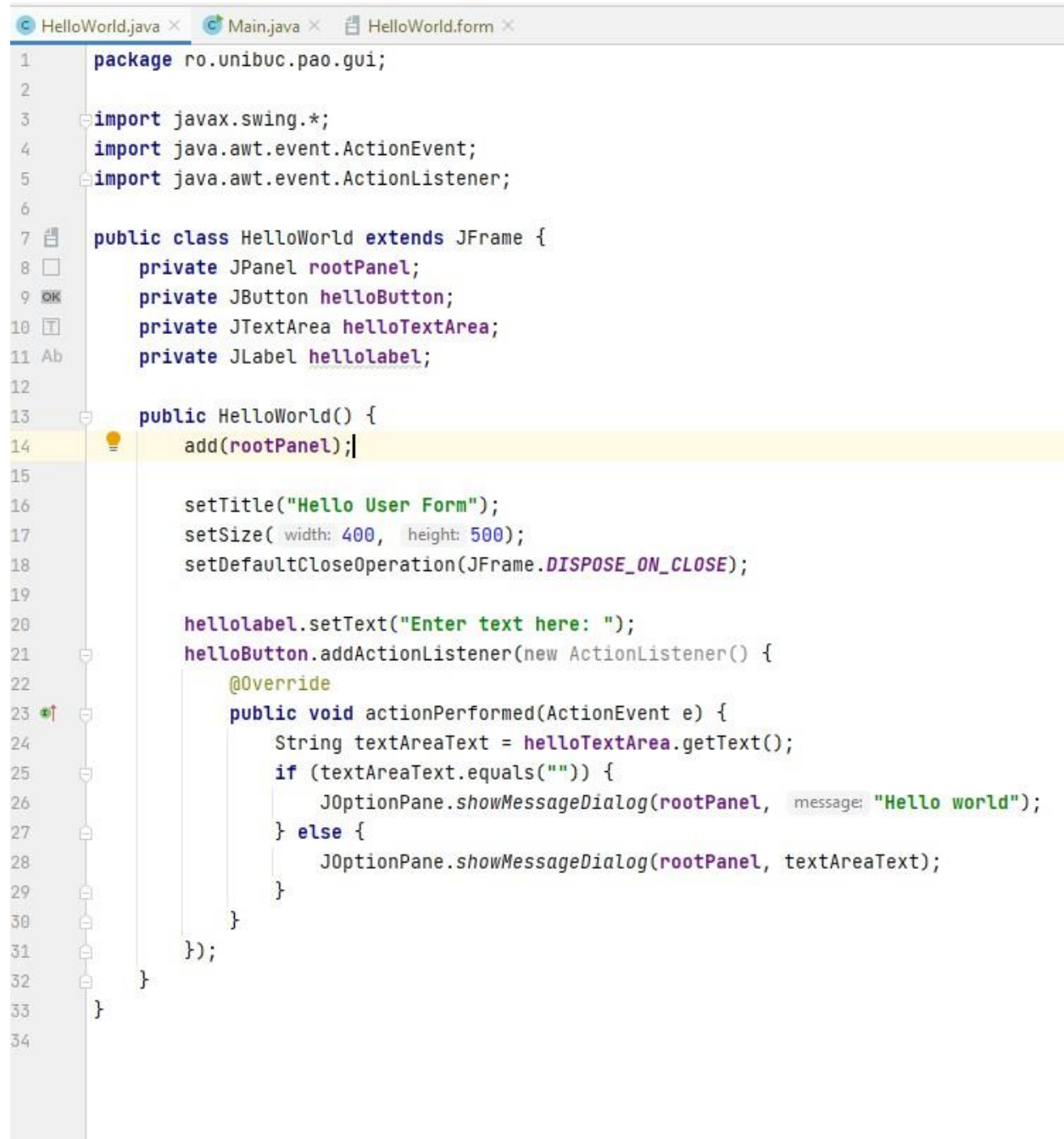6. Ensure that the application runs the form on startup:

```java
public static void main(String[] args) {
JFrame jFrame = new JFrame("Hello World");
    jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    HelloWorld jPanel = new HelloWorld();
jFrame.setContentPane(jPanel.getRootPanel());

    jFrame.pack();
jFrame.setVisible(true);
}
```

7. When running the code, as the JPanel has no content, you will only see the title bar of an application, with no contents.
8. Next, drag a JTextArea onto it and set it to have a field name of helloTextArea.
9. Add a JButton below the label and set its field name to helloButton. Right-click on the button and choose 'Create Listener' from the menu, then from the dialogue box that appears, choose 'ActionListener'. This creates a method in the code which will be called when a user clicks on the button.
10. You can get the text entered in the text area my calling the *getText* method and display it on a message dialog when the button is pressed: *JOptionPane.showMessageDialog*
11. We can expand the size of the box by calling setSize method, change the layout by calling changeLayout (FlowLayout, BorderLayout, GridLayout, etc), or the title by calling setTitle.
12. We can remove the code in Main class and remain only with the below if our HelloWorld class would extend JFrame and move in constructor all the set method called before in main:

```java
public static void main(String[] args) {
HelloWorld jFrame = new HelloWorld();
    jFrame.setVisible(true); // doar asa devine vizibil container
}
```

HelloWorld.java ×   Main.java ×   HelloWorld.form ×

```java
package ro.unibuc.pao.gui;

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

public class HelloWorld extends JFrame {
    private JPanel rootPanel;
    private JButton helloButton;
    private JTextArea helloTextArea;
    private JLabel hellolabel;

    public HelloWorld() {
        add(rootPanel);

        setTitle("Hello User Form");
        setSize( width: 400,   height: 500);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        hellolabel.setText("Enter text here: ");
        helloButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String textAreaText = helloTextArea.getText();
                if (textAreaText.equals("")) {
                    JOptionPane.showMessageDialog(rootPanel,  message: "Hello world");
                } else {
                    JOptionPane.showMessageDialog(rootPanel, textAreaText);
                }
            }
        });
    }
}
```