Theory: The basic window in Swing

7 minutes 0 / 5 problems solved

§1. What is Swing?

Swing is a lightweight **GUI (Graphical User Interface)** library that is used to create **window-based** Java applications. This library eases the Java developer's life by providing a lot of graphical components (such as buttons, labels, and so on).

Swing's benefits include:

- the high degree of customization – it is possible to modify any UI component according to your needs;
- platform-independence – unlike its predecessor called **AWT**, the **Swing** library allows you to write GUI once and run it on any OS with no trouble.

We hope you are interested in this library now. It's time to write and run a simple program.

§2. Hello World-ing Swing

In Swing, all of the classes that represent graphic elements start with the letter `J`. They are located in the `javax.swing` package.

The `JFrame` class represents **an empty window**. To improve the basic window, you should extend this class and then customize the subclass as you need.

Here is the `HelloFrame` class that demonstrates the window with the predefined size and a title.

```
import javax.swing.*;

public class HelloFrame extends JFrame {

    public HelloFrame() {
        super("My First Swing App");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 300);
        setVisible(true);

    }



    public static void main(String[] args) throws Exception {

        Runnable initFrame = new Runnable() {

            @Override

            public void run() {

                new HelloFrame();

            }

        };



        SwingUtilities.invokeAndWait(initFrame);

    }

}
```

Current topic:

The basic window in Swing

Topic depends on:

Libraries

Inheritance Stage 7

Topic is required for:

Swing components

We create an instance of `Runnable` and pass it into `SwingUtilities.invokeAndWait` method to initialize a frame. Further, we will discuss why we need to initialize Swing applications in that way.

There are some explanations about how it works.

The `HelloFrame` class extends `JFrame` and customizes its default properties. The constructor sets the title `"My First Swing App"` as the window's title, the size (in pixels), and the visibility of the windows.

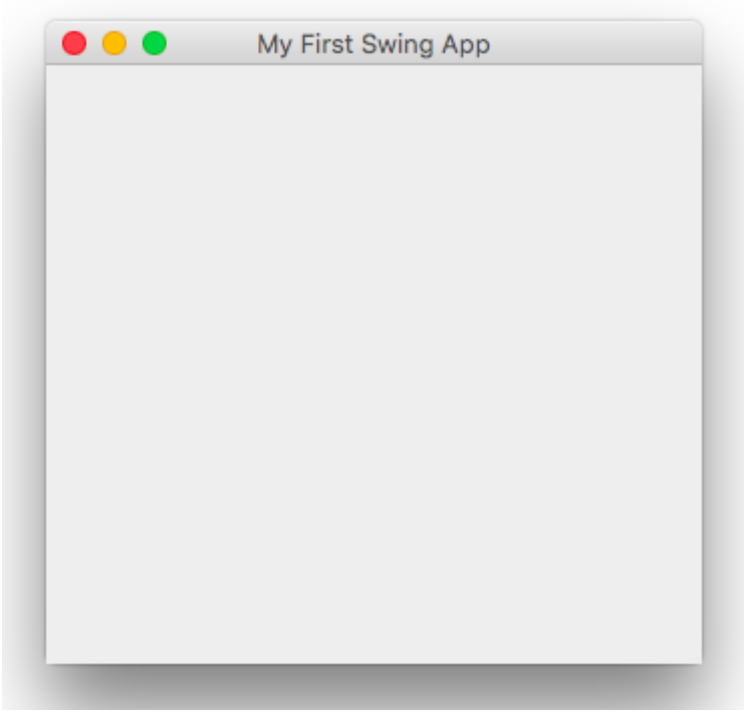
The `setDefaultCloseOperation` method inherited from `JFrame` sets the default close operation. Here, the constant `JFrame.EXIT_ON_CLOSE` becomes the default operation after the close button on the title bar is clicked on. Instead of being hidden, the `JFrame` is disposed of when the frame is closed. It allows the JVM to exit and the program to terminate.

We've reached the smallest part that needs a thorough explanation. The `main` method just creates the frame to be displayed. Once it has been started, the AWT event dispatching thread keeps working until all of the Swing top-level windows are disposed of. The program will not stop even after the completion of all the code in the `main` method.

§3. Enjoying the window

Once we have created a window, it is time to start the program and enjoy it!

> The window looks a little different on other operating systems, but its behavior is always the same.



The created window (on MacOS)

This window has a size of 300 x 300 pixels. It can be closed by clicking the standard closing button (it depends on the OS).

In the next topics, we will learn how to add other graphical components (such as buttons, labels, and so on) on the top of the window.

A few questions for this step to make sure you got the intro right and we'll dive into components.

Report a typo

185 users liked this theory. **3** didn't like it. **What about you?**

Start practicing

Comments (6) Hints (0) Useful links (1) Show discussion