Swing

Lukáš Sedlák, Aleš Hrabě

November 19, 2012

What is Swing?

- Sun's GUI library for the Java
- part of Oracle's Java Foundation Classes
- Model-View-Controller architecture
- replaces Abstract Window Toolkit AWT
- written entirely in Java
- part of the Java Standard Edition
- composed of 18 packages
- main package javax.swing

Swing vs AWT

- AWT is faster than Swing
- AWT applets do not need web plugin
- AWT uses platform API's
- AWT has platform specific limitations
- AWT has less features
- AWT has less support from 3rd parties
- Swing has own painting = no DirectX
- AWT for simple app on spec.platform

Hello World: class

```
package start;
import javax.swing.*;
public class HelloWorldSwing {
    private static void HelloWorld() { ... }
    public static void main(...) {... }
}
```

Hello World: main

```
public static void main(String[] args) {
    javax.swing.SwingUtilities.invokeLater(
    new Runnable() {
        public void run() {
            HelloWorld();}
    } );
}
```

Hello World: helloworld

```
private static void HelloWorld() {
    JFrame frame = new JFrame("Hi World!");
    frame.setDefaultCloseOperation
        (JFrame.EXIT_ON_CLOSE);
    frame.pack();
    frame.setVisible(true);
}
```

Compilation

Debian package default-jdk

http://www.java.com

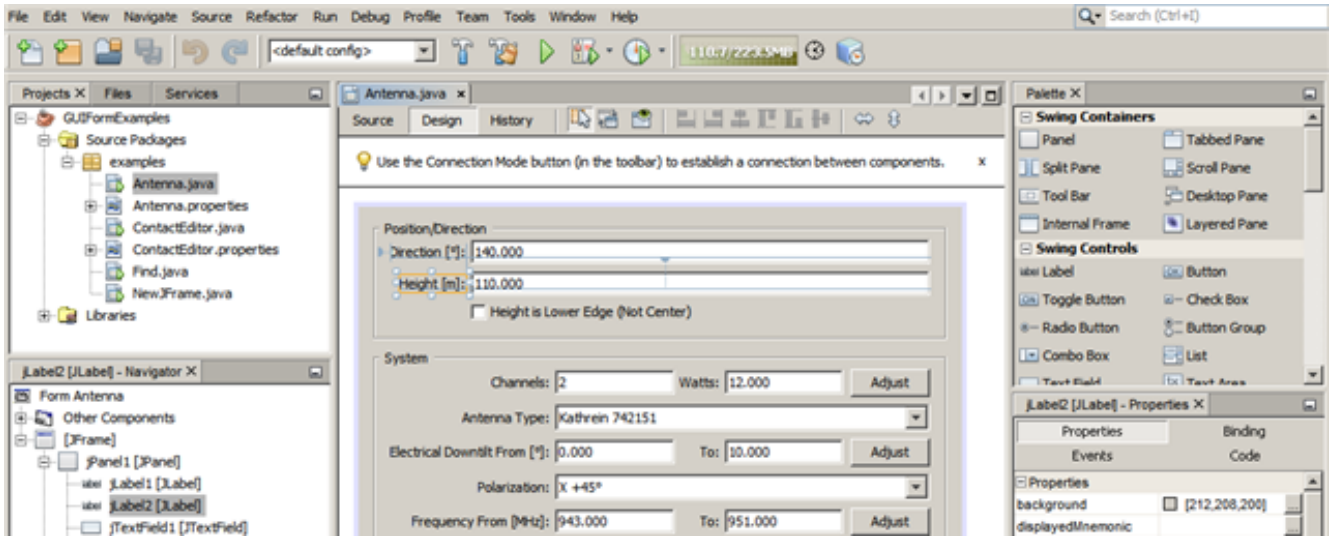
http://openjdk.java.net

```
$ javac start/HelloWorldSwing.java
$ java start.HelloWorldSwing
```

Swing IDEs

SpeedJG

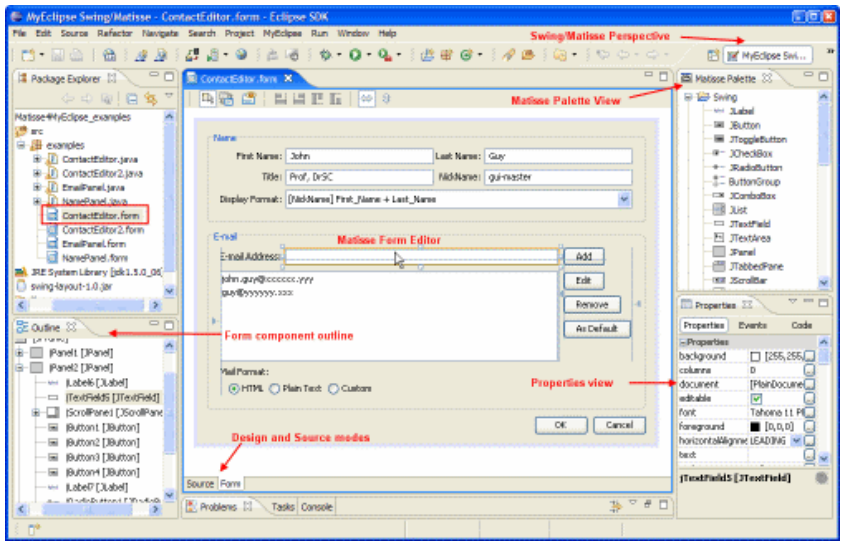
NetBeans Swing IDE



Swing IDEs

Window Builder Pro for Eclipse

Jigloo for Eclipse



Label example

```
private static void HelloWorld() {
    ... frame creation

    JLabel label = new JLabel("Hi World");
    frame.getContentPane().add(label);

    ... size, visibility
}
```

Button example 1

```
private static void HelloWorld() {
    ... frame creation

    JButton button = new JButton("Zzz");
    frame.add(button);

    ... size, visibility
}
```

Button example 2

```
package start;

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

public class HelloWorldSwing3 {
    ... }
```

Button example 2

```
public static void main(String[] args) {
   EventQueue.invokeLater(new Runnable(){
      public void run(){
         new HelloWorldSwing3().HelloButton();
      }
   });
}
```

Button example 2

```
private void HelloButton(){
  ...
  button.addActionListener(
    new ActionListener() {
      public void actionPerformed
                              (ActionEvent e) {
        button.setText("Whats up?");
} }); }
```

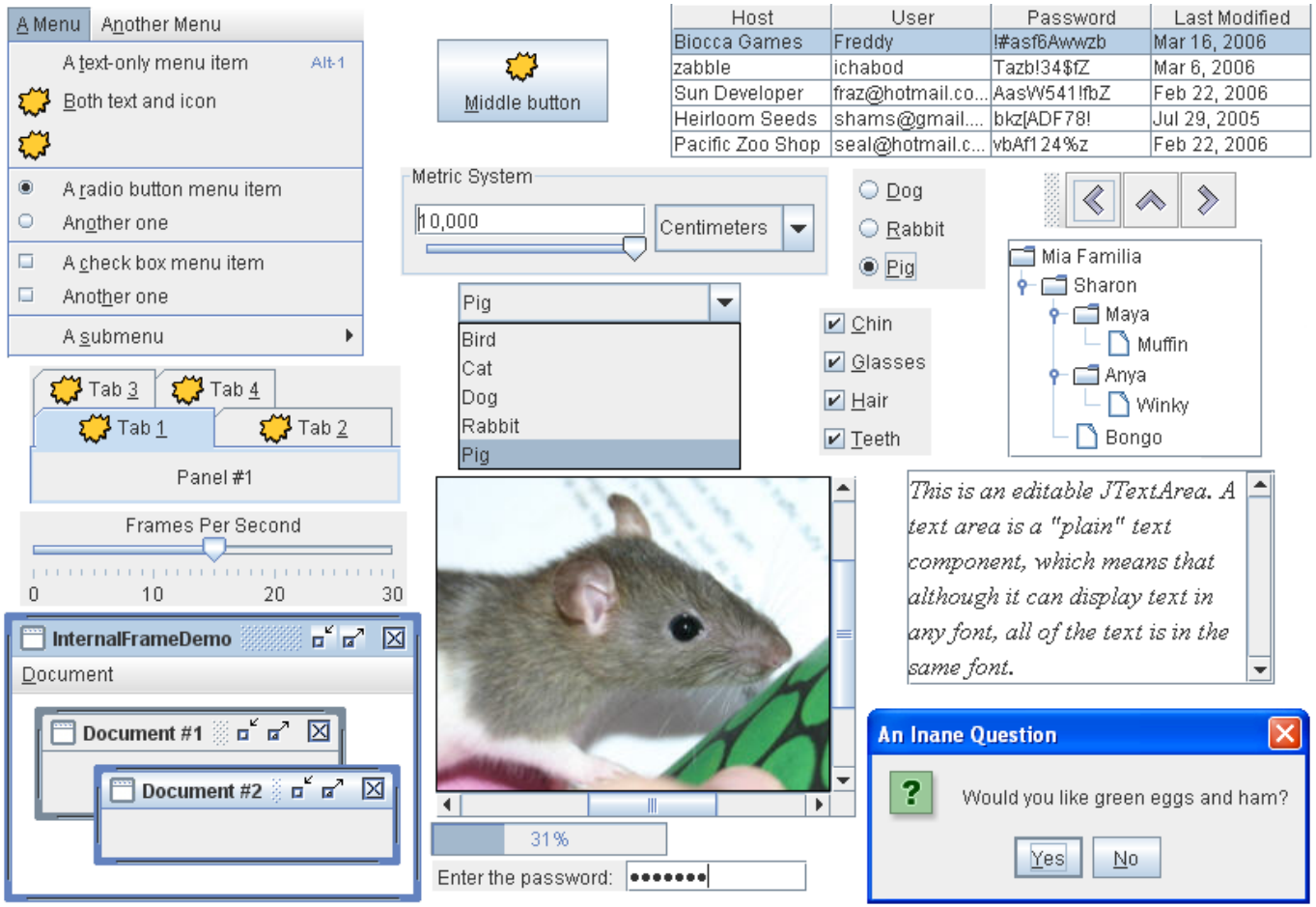
Text box example

```
...
final JTextField mytextfield =
                        new JTextField(20);
final JLabel mylabel = new JLabel();
...
frame.add(mytextfield);
frame.add(mylabel);
```

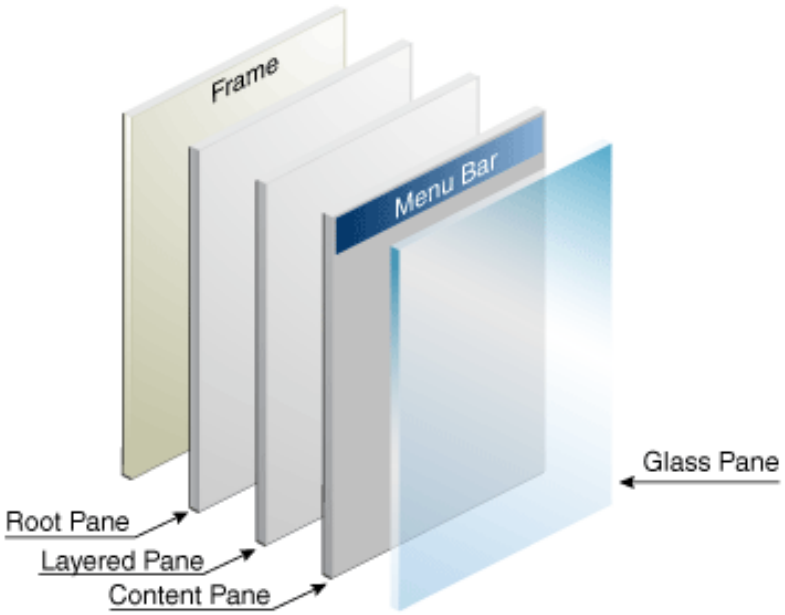
Text box example

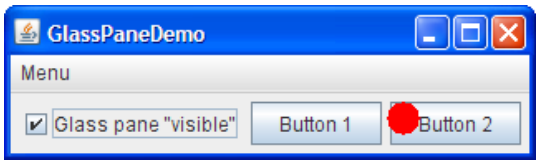
```
...
mytextfield.addActionListener(
                    new ActionListener() {
  public void actionPerformed(
                      ActionEvent e) {
    String text = mytextfield.getText();
    mylabel.setText(text);
    mytextfield.selectAll(); } });
```

Controls overview

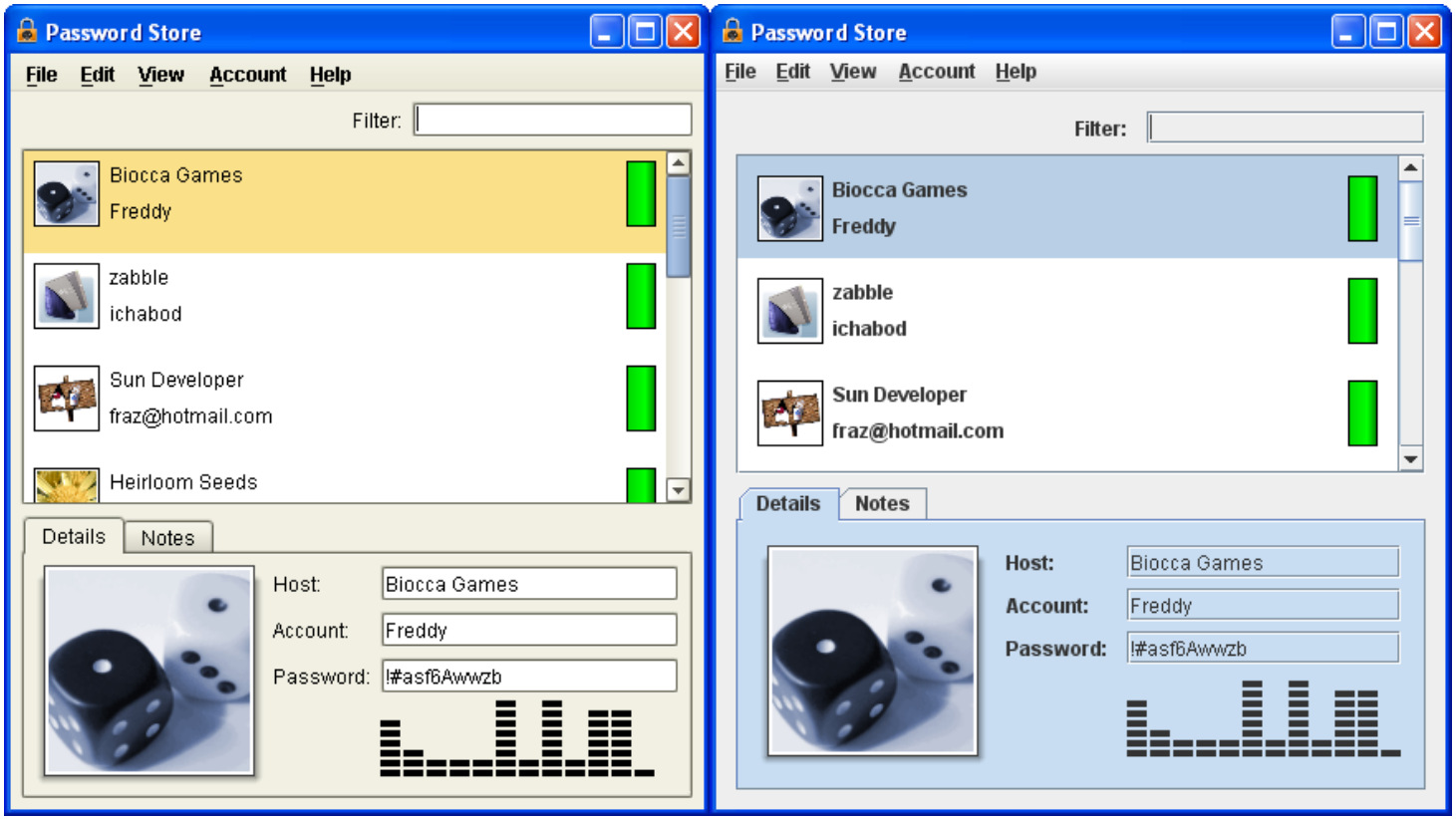


Conteiners overview

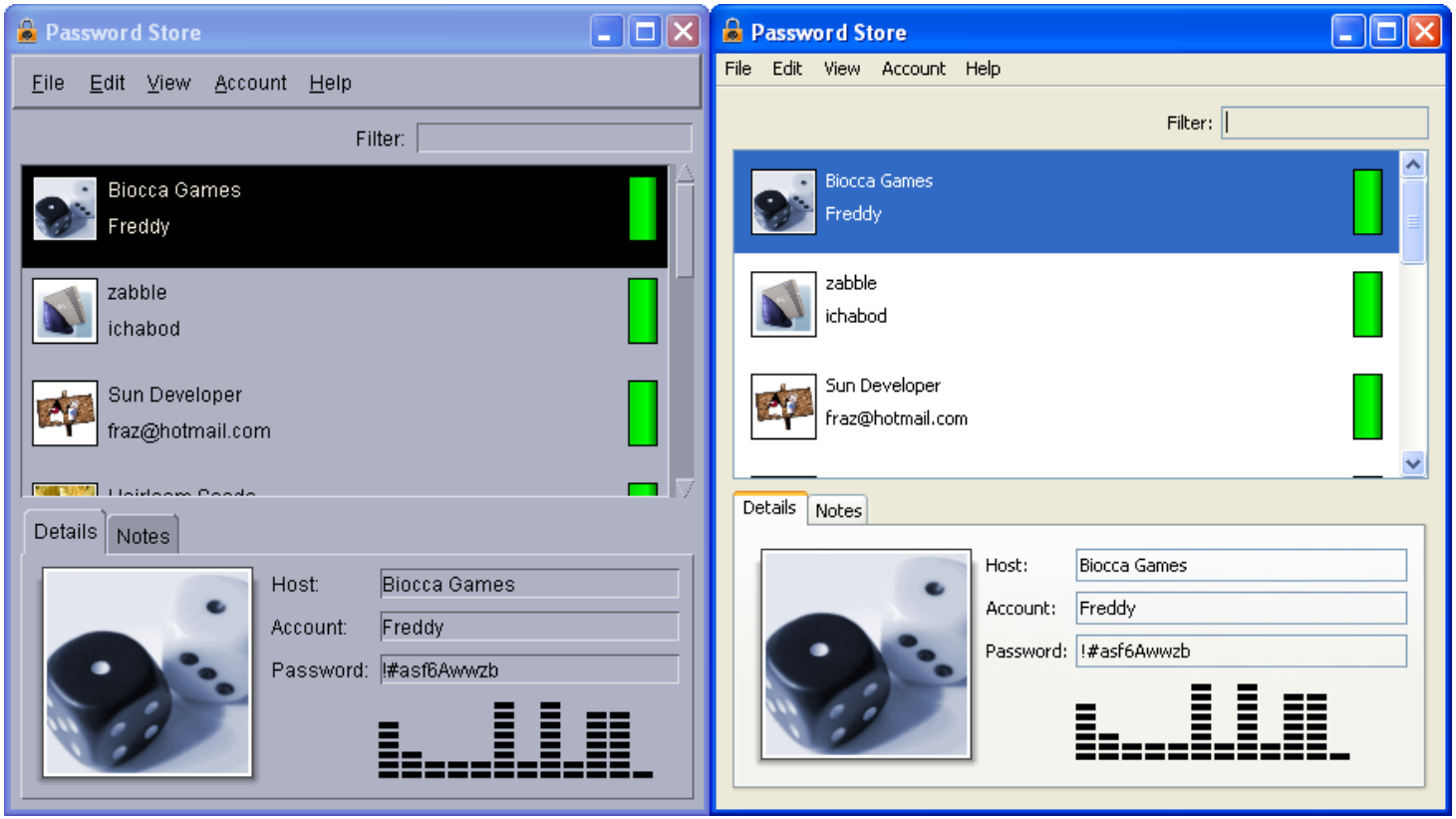




Themes overview



Themes overview



Other abilities

data transfer between various programs

drag and drop, cut, copy, paste possible

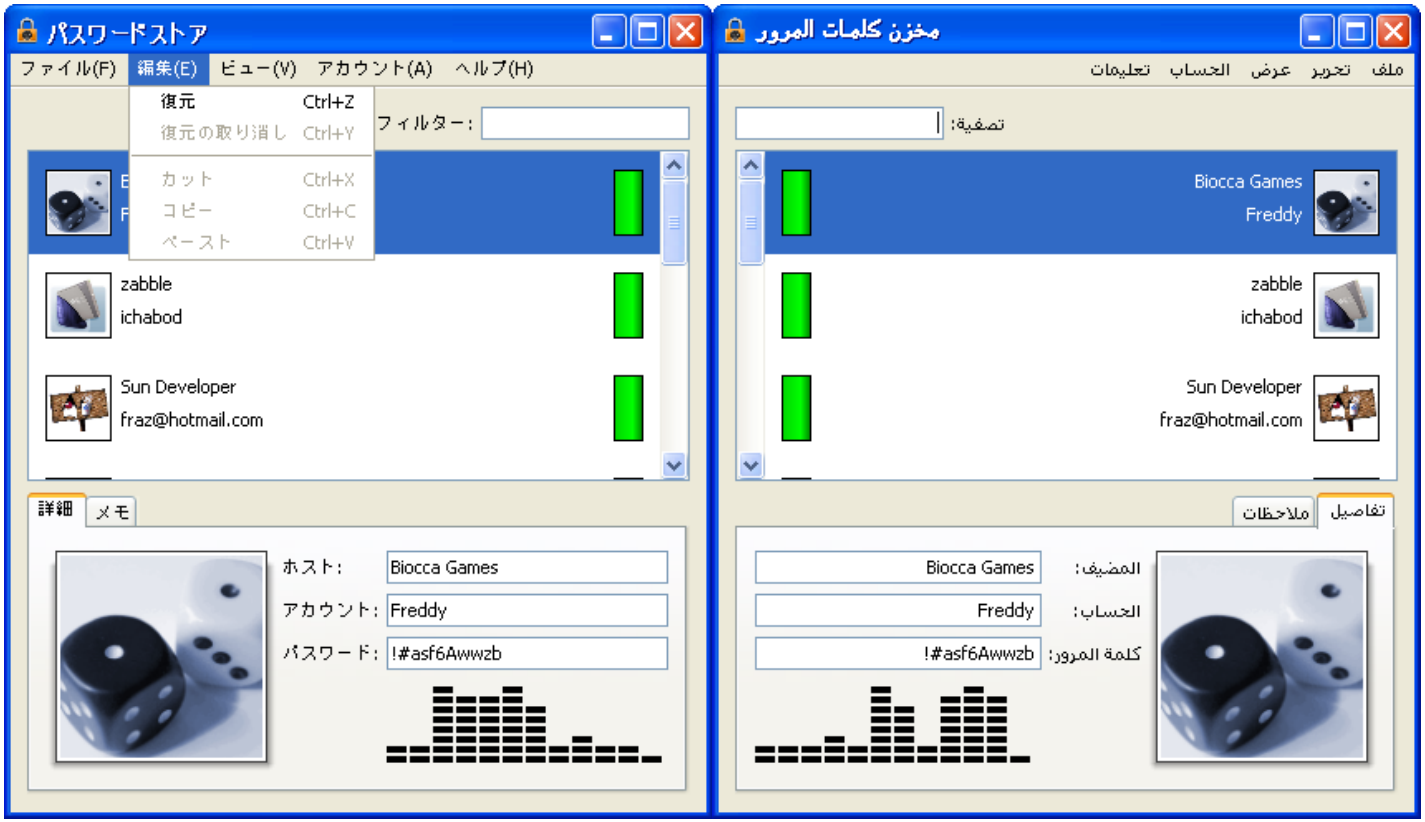
internationalization = no recompilation

texts stored outside code

localization = translating text

various text alignment and flows

Internationalization



Other abilities 2

voice interface, keyboard enhancement

Java Access Bridge for java-windows

- interface for assistive technologies

Java Accessibility Utilities

Desktop API for integrating GUI

- hypertext links, email clients, printing

system tray

Concurrency

for never-freezing GUI
initial thread from the application
event dispatch thread for event-handling
backgroud thread for time-consuming tasks
threads inter-communication needed

Initial thread

main, init, start, etc.

creates a Runnable object

    for GUI initialization

schedules the object for execution

    on event dispatch thread

javax.swing.SwingUtilities.invokeLater

javax.swing.SwingUtilities.invokeAndWait

Event dispatch thread

special thread for handling events

Swing object methods are not thread safe

thread interference possible

memory consistency errors possible

Swing object methods invoked from the thread

thread-safe GUI library impossible

series of short tasks

javax.swing.SwingUtil.isEventDispatchThread

Background thread

special thread for long tasks

each thread represented by instance

javax.swing.SwingWorker

method "done" send to event thread

cancellation, partial results possible

input parameter changes possible

loading graphics, working with arrays

timeout option

Background thread

SwingWorker.publish, SwingWorker.process
SwingWorker.cancel for stopping process
SwingWorker.isCancelled often testing
bound properties: "progress" and "state"
    for inter-thread communication
SwingWorker.setProgress, getProgress, 0-100
SwingWorker.getState, PENDING,
    STARTED, DONE