Java Swing

What is Java Swing?

- Part of the Java Foundation Classes (JFC)
- Provides a rich set of GUI components
- Used to create a Java program with a graphical user interface (GUI)
- table controls, list controls, tree controls, buttons, and labels, and so on…

What features are available?

- GUI components like button, checkbox, and so on…
- Java 2D API: images, figures, animation
- Pluggable look and feel: use samples or create your own
- Data Transfer: cut, copy, paste, drag & drop
- Internationalization: supports different input language, right to left reading
- Accessibility API: for people with disabilities
- Undo Framework API: supports unlimited numbers of actions to undo and redo
- Flexible Deployment: run within a browser as an applet or Java Web Start

How does HelloWorld look like?

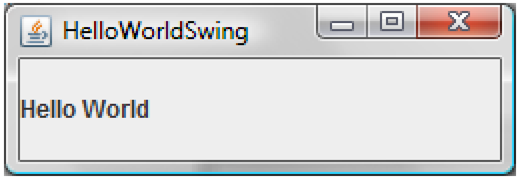
```
import javax.swing.*;

public class HelloWorldSwing {
  private static void createAndShowGUI() {
    //Create and set up the window.
    JFrame frame = new JFrame("HelloWorldSwing");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //Add the ubiquitous "Hello World" label.
    JLabel label = new JLabel("Hello World");
    frame.getContentPane().add(label);

    //Display the window.
    frame.pack();
    frame.setVisible(true);
  }

  public static void main(String[] args) {
    //Schedule a job for the event-dispatching thread:
    //creating and showing this application's GUI.
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        createAndShowGUI();
      }
    });
  }
}
```



Hello World

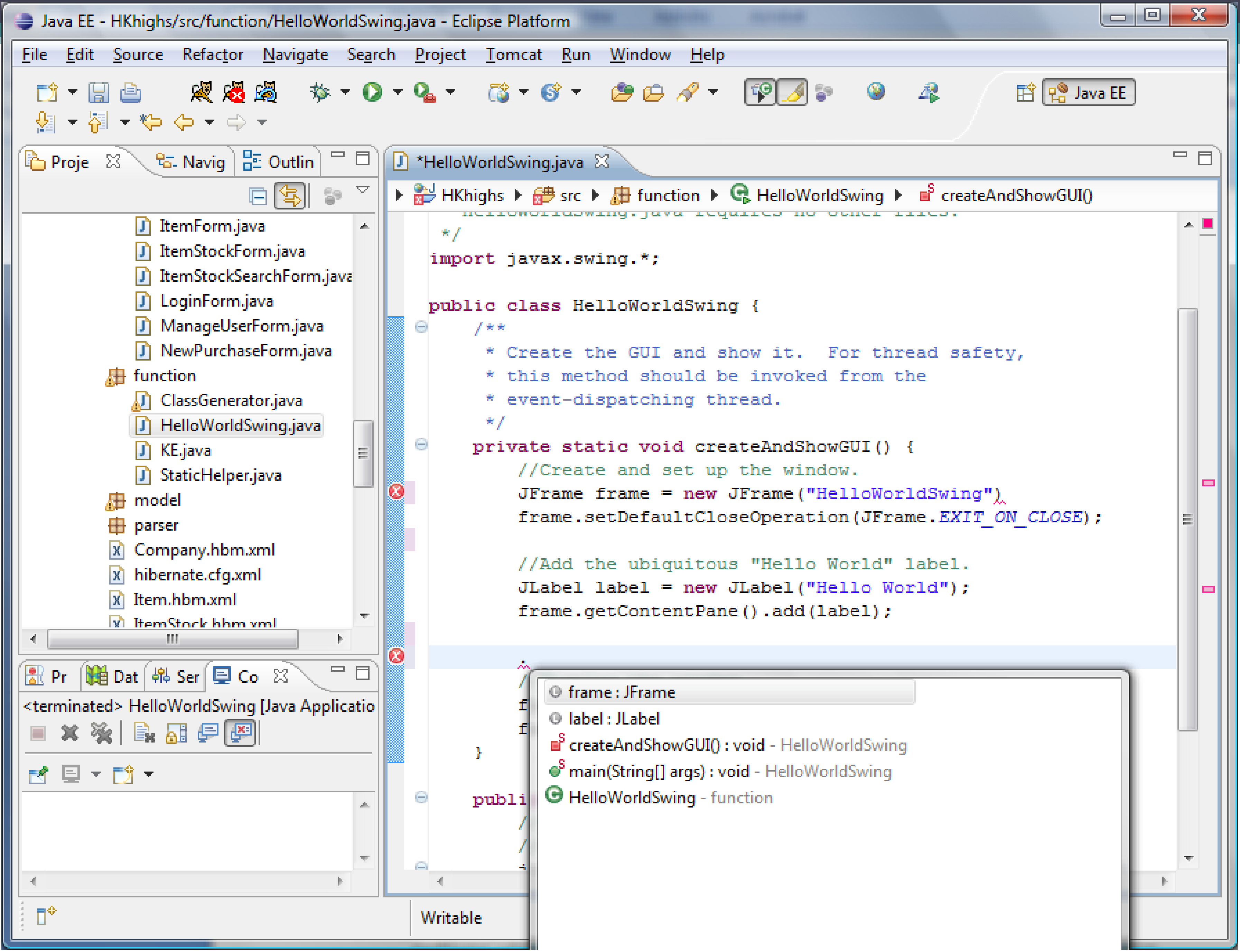
Who are users?

- Since we are evaluating user interface toolkit itself, in this case Java Swing, users will be software developers, not software users
- I believe that most Java developers use Eclipse as their developing platform so we will evaluate Java Swing with Eclipse based on ten usability heuristics by **Jakob Nielson**

1. Visibility of system status

- This may be a **strong advantage of Java Swing** over other UI toolkits, not because of Java Swing itself is great, but because **Eclipse** provides such sophisticated checking on what is going on now
- Constantly checks for syntax errors
- Lists available methods or variables when **press '.'** (dot)

- You will have to run it in order to see the status of your program



<terminated> HelloWorldSwing [Java Applicatio

```
   HelloWorldSwing.java requires no other files.
 */
import javax.swing.*;

public class HelloWorldSwing {
    /**
     * Create the GUI and show it.  For thread safety,
     * this method should be invoked from the
     * event-dispatching thread.
     */
    private static void createAndShowGUI() {
        //Create and set up the window.
        JFrame frame = new JFrame("HelloWorldSwing")
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //Add the ubiquitous "Hello World" label.
        JLabel label = new JLabel("Hello World");
        frame.getContentPane().add(label);

        .
        /
        f
        f
    }

    publi
        /
        /
```

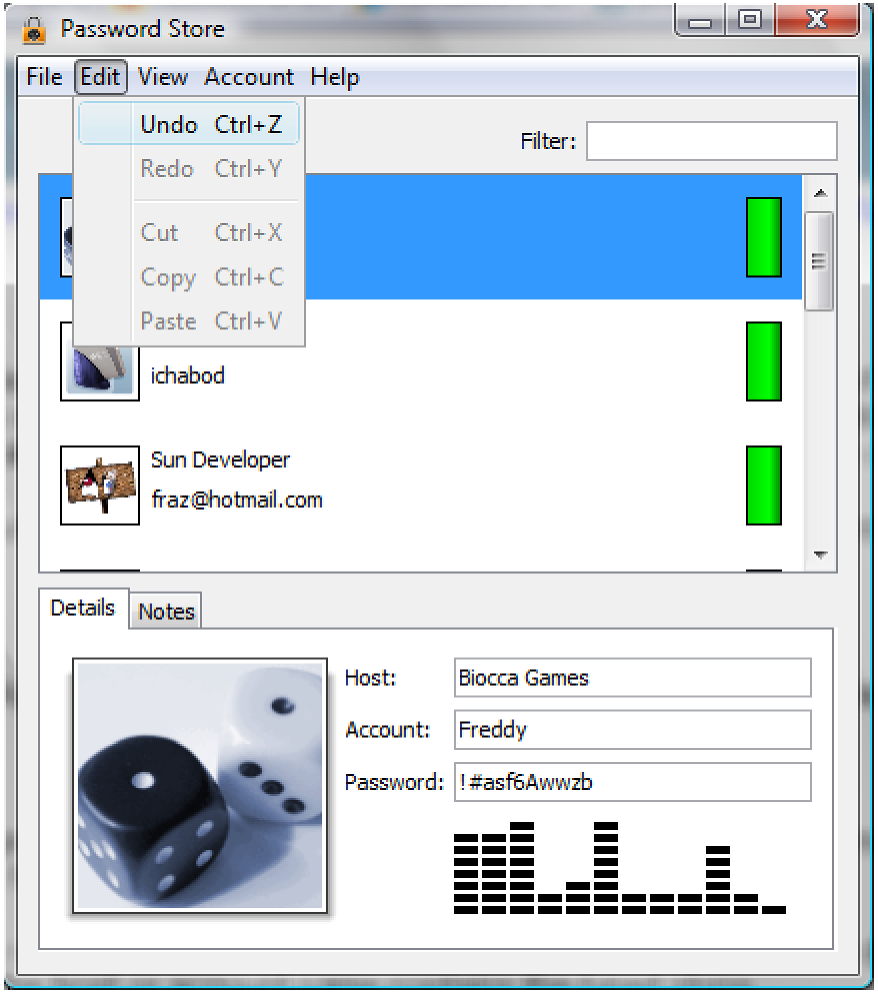
```
frame : JFrame
label : JLabel
createAndShowGUI() : void - HelloWorldSwing
main(String[] args) : void - HelloWorldSwing
HelloWorldSwing - function
```

2. Match between system and the real world

- First of all, it is Java
- It follows Java convention
- It consists of 18 public packages of Java classes
- Its classes and methods are reasonably named
- Unless you are the first time programmer, you don’t have to worry about its syntax or convention
-
  ```
  JLabel developerNameLabel = new javax.swing.JLabel();
  developerNameLabel.setFont(new java.awt.Font("Arial", 0, 14));
  developerNameLabel.setForeground(new java.awt.Color(255, 255, 255));
  ```

3. User control and freedom

- Eclipse supports strong undo and redo features
- You can't possibly go wrong and have to rewrite every code
- You can always fix it even though it may take you some effort and time
- Java Swing also provides undo and redo package
- javax.swing.event.UndoableEditEvent;
  javax.swing.event.UndoableEditListener;
  javax.swing.undo.UndoableEdit;



ichabod

Sun Developer
fraz@hotmail.com

Host: Biocca Games

Account: Freddy

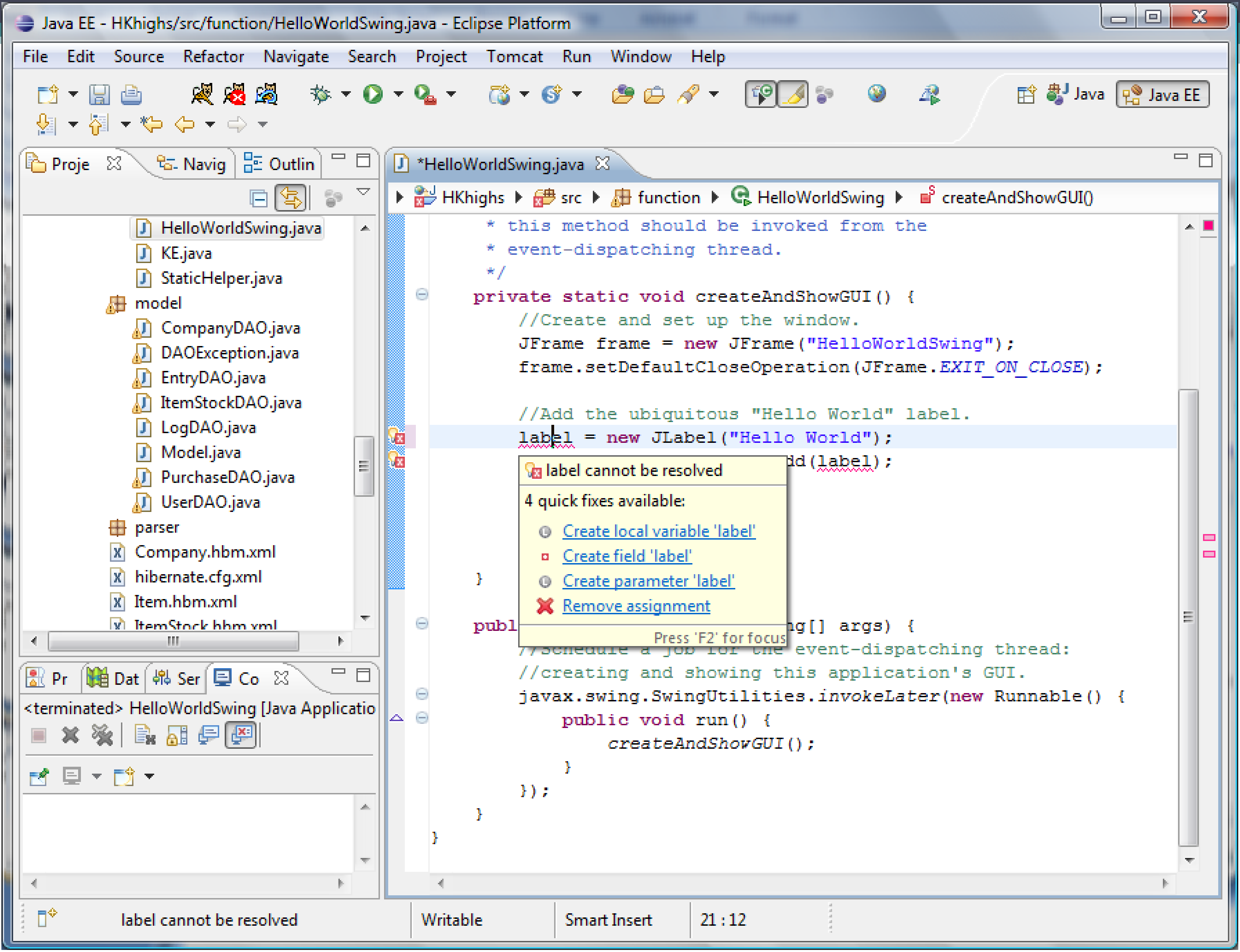
Password: !#asf6Awwzb

4. Consistency and standards

- Java Swing follows Java convention
- Packages, classes, methods, parameters, variables
- Mainly constructor, getter, setter

5. Error prevention

- First of all, Java is a strongly typed language: primitives and objects
- Eclipse checks for syntax and type errors continuously
- It gives red underline to errors and small red box to let you know which line in document it is located
- If you move your mouse over the error, then it suggests possible solutions to the error
- If you think one of the suggestions is a solution, then simply click on it to apply it
- Of course, it won’t compile and run until you fix all the syntax errors
- However, you don’t have any idea about runtime errors except that you will have to run it and find out



```
     * this method should be invoked from the
     * event-dispatching thread.
     */
    private static void createAndShowGUI() {
        //Create and set up the window.
        JFrame frame = new JFrame("HelloWorldSwing");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //Add the ubiquitous "Hello World" label.
        label = new JLabel("Hello World");
                                    dd(label);




    }

    publ                                    ng[] args) {
        //Schedule a job for the event-dispatching thread:
        //creating and showing this application's GUI.
        javax.swing.SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }
}
```

label cannot be resolved

4 quick fixes available:

- Create local variable 'label'
- Create field 'label'
- Create parameter 'label'
- Remove assignment

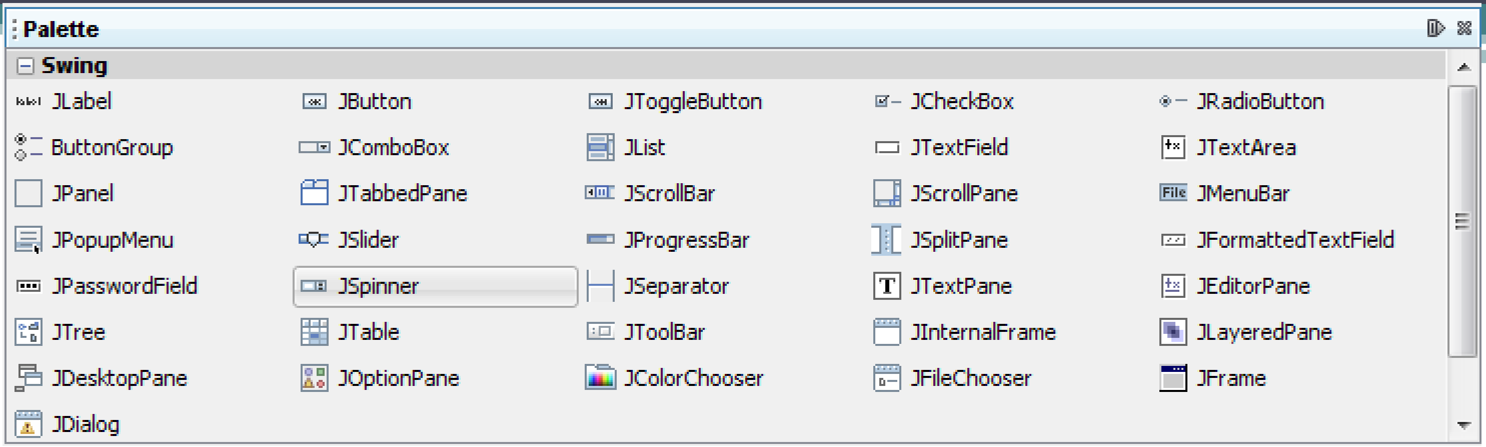
Press 'F2' for focus

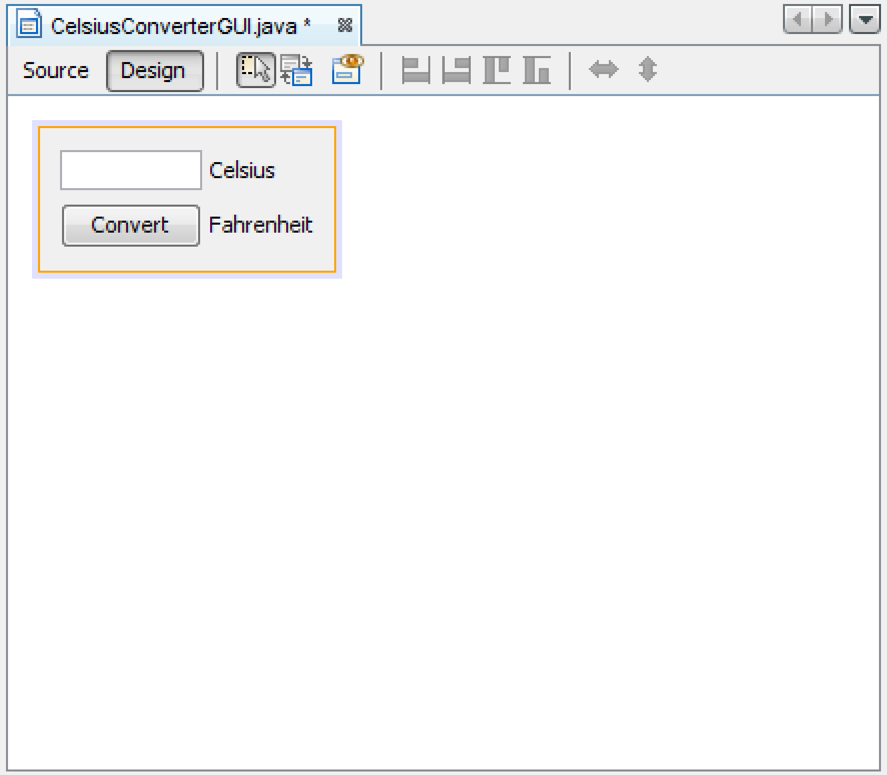
<terminated> HelloWorldSwing [Java Applicatio

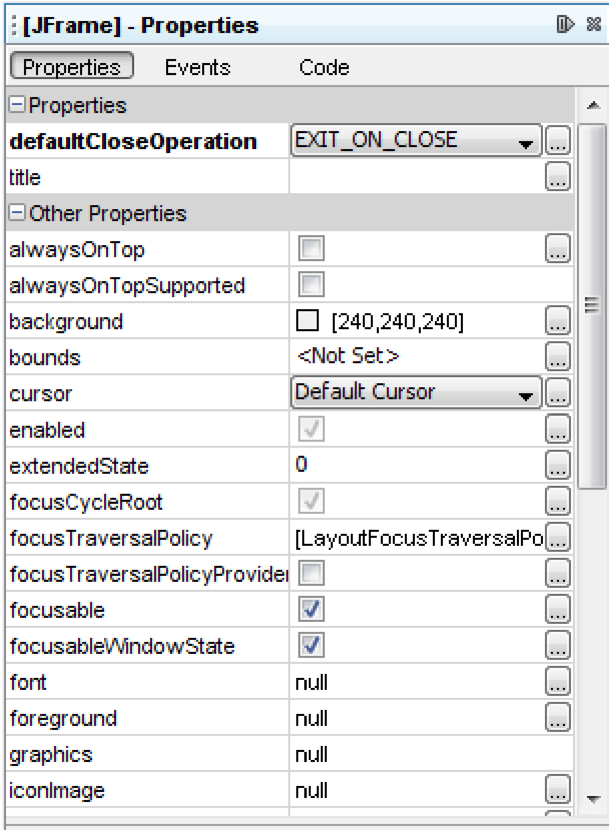
label cannot be resolved | Writable | Smart Insert | 21 : 12

6. Recognition rather than recall

- It’s a programming language, so it’s better and faster for you if you memorize names and their functions of classes or methods
- However, whenever you **press ‘dot’** after name of package, class, or object, then eclipse provides you a list of all possible subclasses, functions and variables
- If you move your mouse over almost anything, then eclipse provides you with a text document associated with it, usually javadoc, or you can click on it and it directs you to online javadoc page
- You don’t have a help of graphical interface to develop a graphical interface, so it maybe a disadvantage of Java Swing with eclipse.
- By the way, you can have a graphical interface if you use NetBeans IDE instead of eclipse







⊟ Properties

defaultCloseOperation	EXIT_ON_CLOSE
title	

⊟ Other Properties

alwaysOnTop	☐
alwaysOnTopSupported	☐
background	□ [240,240,240]
bounds	<Not Set>
cursor	Default Cursor
enabled	☑
extendedState	0
focusCycleRoot	☑
focusTraversalPolicy	[LayoutFocusTraversalPo
focusTraversalPolicyProvider	☐
focusable	☑
focusableWindowState	☑
font	null
foreground	null
graphics	null
iconImage	null

7. Flexibility and efficiency of use

- Swing includes many basic components as a package, so it is efficient to use them
- At the same time, you can create almost anything you want as combination of those components and some pure coding in Java
- Java have had a reputation for being slower and requiring more memory than those written in natively compiled languages such as C or C++
- However, the performance heavily depends on how you optimize your codes and which components of UI you use the most frequently
- It may be subsequently slower or faster

8. Aesthetic and minimalist design

- Swing is designed in a way that it provides a set of "lightweight" (all-Java language) components that, to the maximum degree possible, work the same on all platforms
- It includes almost only and all components we could find around any software with user interface
- Yet, it gives developers varieties to customize those components

9. Help users recognize, diagnose, and recover from errors

- Syntax and type checking errors are already covered previously
- Java shows where in the code runtime errors (exceptions) are caused, specifying line numbers and brief reason for error on the console of eclipse
- It’s not always right, but most of the times it is right
- It’s relatively easy to find the cause of the error and debug it comparing to other languages I have experienced with

10. Help and documentation

- Javadoc
- Eclipse’s support for javadoc (already covered)

Conclusion

- Java Swing is easier to learn than others because it’s Java
- You can use any helpful tools out there that are for Java development like eclipse IDE, NetBeans IDE
- Lacks live graphical and interactive help while developing
- Has unlimited possibilities depending on how you implement your software