Graphical User Interface using widgets

Object Oriented Programming
2022 First Semester

Shin-chi Tadaki (Saga University)

1. GUI in Java
2. `java.awt`
3. `javax.swing`
4. Working with `JFrame`
5. GUI without actions

GUI (Graphical User Interface) in Java

- GUI libraries in general
  - X11 with c/c++, etc.
  - OS dependent
- in Java
  - GUI libraries are distributed with JDK
  - OS independent
  - Working under OS dependent window managers

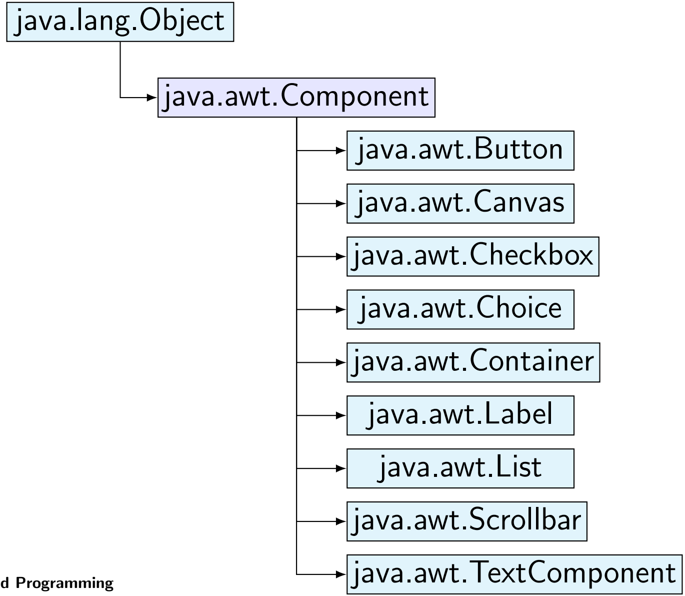
GUI programming as OOP

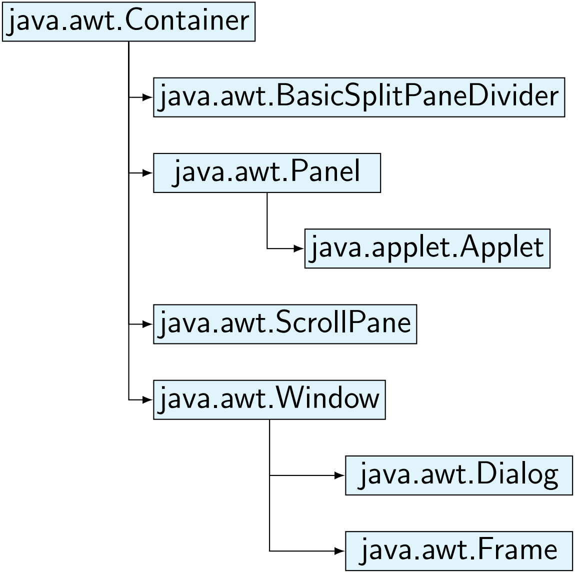
- GUI uses various widgets (window gadgets)
- Each widget has own properties and methods
  - Properties: color, size, etc.
  - Methods: action, property change, visible, etc.
- Fundamental widgets are used for applications by extensions.
  - GUI applications by extending `JFrame`
  - Widget containers by extending `JPanel`

Abstract Windows Toolkit java.awt

- Fundamental graphical properties
  - Color, BasicStroke, Font, etc.
- Fundamental widgets
  - panels, buttons, etc.
- Fundamental events
  - mouse, keyboard, property changes, etc.

Hieralchy of java.awt




java.awt.Container

java.awt.BasicSplitPaneDivider

java.awt.Panel

java.applet.Applet

java.awt.ScrollPane

java.awt.Window

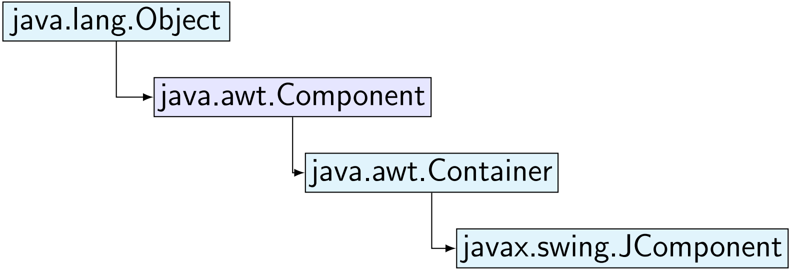
java.awt.Dialog

java.awt.Frame

javax.swing

- Extensions of java.awt
- Enriching widgets
- Complete OS independence
    - Control under OS window manager
    - Separate Look-and-Feel
- Lightweight
- Running as threads

Hieralchy of swing widgets

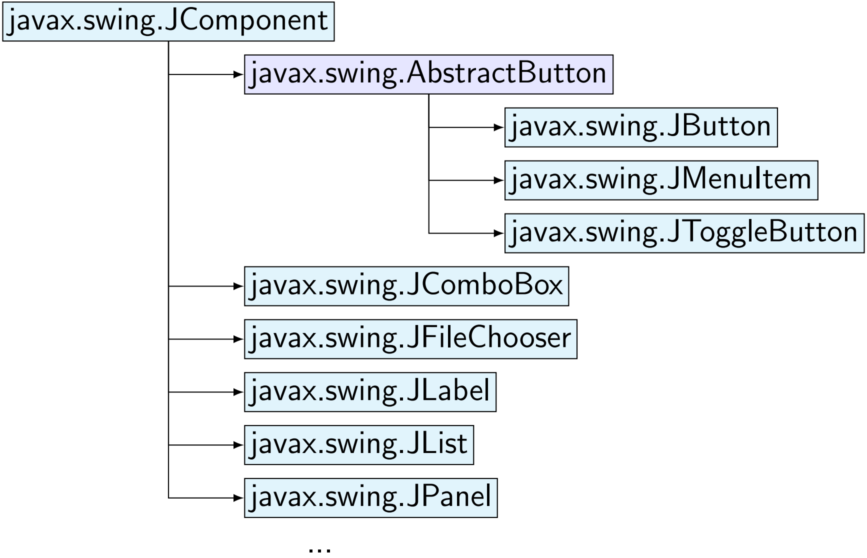


java.lang.Object

java.awt.Component

java.awt.Container

javax.swing.JComponent



javax.swing.JComponent

javax.swing.AbstractButton

javax.swing.JButton

javax.swing.JMenuItem

javax.swing.JToggleButton

javax.swing.JComboBox

javax.swing.JFileChooser

javax.swing.JLabel

javax.swing.JList

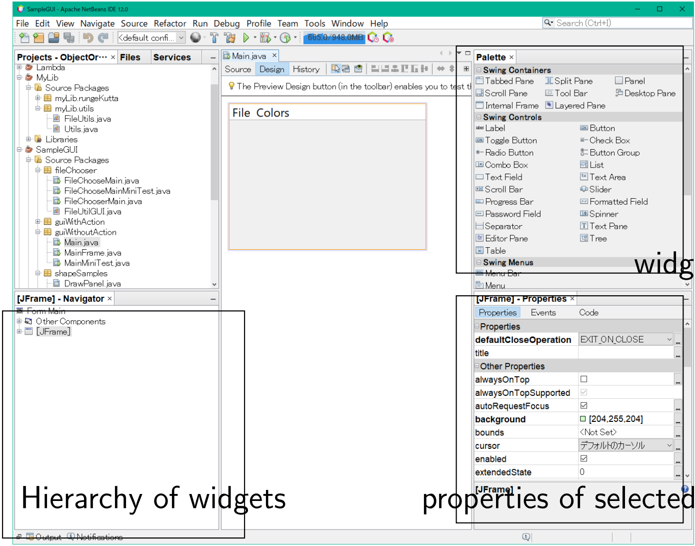
javax.swing.JPanel

...

swing components

- javax.swing.JFrame
  - Main window of applications
  - Put JPanel and JMenuBar onto this component
- javax.swing.JPanel
  - Put widgets on this components
  - Use for drawing
- javax.swing.JMenuBar
  - Menu bar at the top of applications
  - Put javax.swing.JMenu on this component

Layout Design in NetBeans



The Preview Design button (in the toolbar) enables you to test th

Properties

defaultCloseOperation	EXIT_ON_CLOSE
title	

Other Properties

alwaysOnTop	☐
alwaysOnTopSupported	☑
autoRequestFocus	☑
background	🟩 [204,255,204]
bounds	<Not Set>
cursor	デフォルトのカーソル
enabled	☑
extendedState	0

widgets

Hierarchy of widgets

properties of selected widget

Constructing GUI in NetBeans

- Start project as usual cases
- Create `JFrame` form
  - `New`→`JFrame form`
  - At widget hierarchy: `Set Layout`→`BorderLayout`
  - Define a new class as an extension of `JFrame`

Configuring widgets

- Configuring widgets using mouse
  - In `Navigation`: Drag a component from the palette
- creating `JMenuBar`
  - Add `JMenu` and `JMenuItem`

Notice at creating new JFrame instances

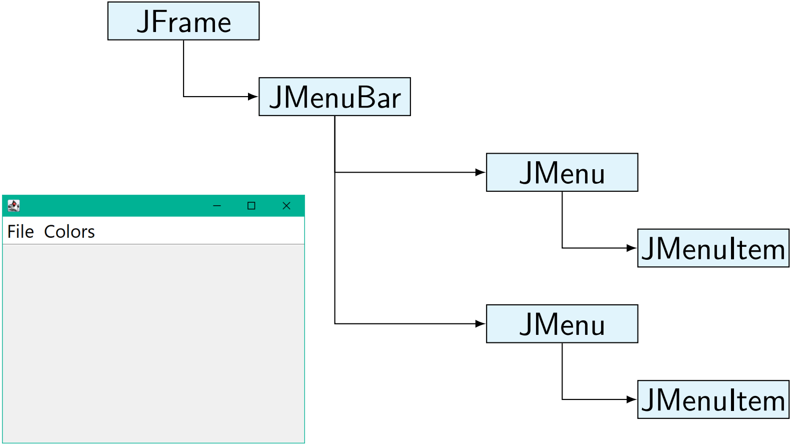
- Properties and layout are stored in `*.form` file.
    - Parts of source files are not allowed to edit, because of stored in `*.form` files.

Today's sample program

https://github.com/oop-mc-saga/SampleGUI

- guiWithoutAction
- guiWithAction
- fileChooser
- simpleTimer

GUI without actions



Two menus in this application

- The first menu `fileMenu`
  - has menu item `exit`
- The second menu `selectColors`
  - has menu items for colors defined in `Colors`
  - Those items are added in the constructor

```java
public class Main extends javax.swing.JFrame {

    public enum Colors {//Define colors as enum instance
        ORANGE(Color.ORANGE), YELLOW(Color.YELLOW), GREEN(Color.GREEN);
        private final Color color;

        Colors(Color color) {  this.color = color;  }

        public Color getColor() {  return color;  }
    }

    public Main() {
        initComponents();
        Font font = new Font("MS UI Gothic", 0, 24);
        for (Colors m : Colors.values()) {
            JMenuItem item = new JMenuItem(m.toString());
            item.setFont(font);
            selectColors.add(item);
        }
    }
    ...
}
```

initComponents()

- Generated automatically with `form` file through NetBeans
- What `initComponents()` does is
  - Inserting widgets and laying out them
  - Setting properties of widgets

Exercise

Add a new menu for selecting color (see quiz).