



Graphical User Interface widgetを使う

オブジェクト指向プログラミング特論

2020年度

只木進一：工学系研究科

JavaとGUI

- 多くのプログラミング言語では
 - GUIは言語とは別のライブラリ
 - 例：c/c++とX11、GTK
 - プラットフォーム依存
- Javaでは
 - GUIライブラリが言語と同封されて配布
 - プラットフォーム独立
 - 各プラットフォームのウィンドウマネージャ利用可

OOPとしてのGUI

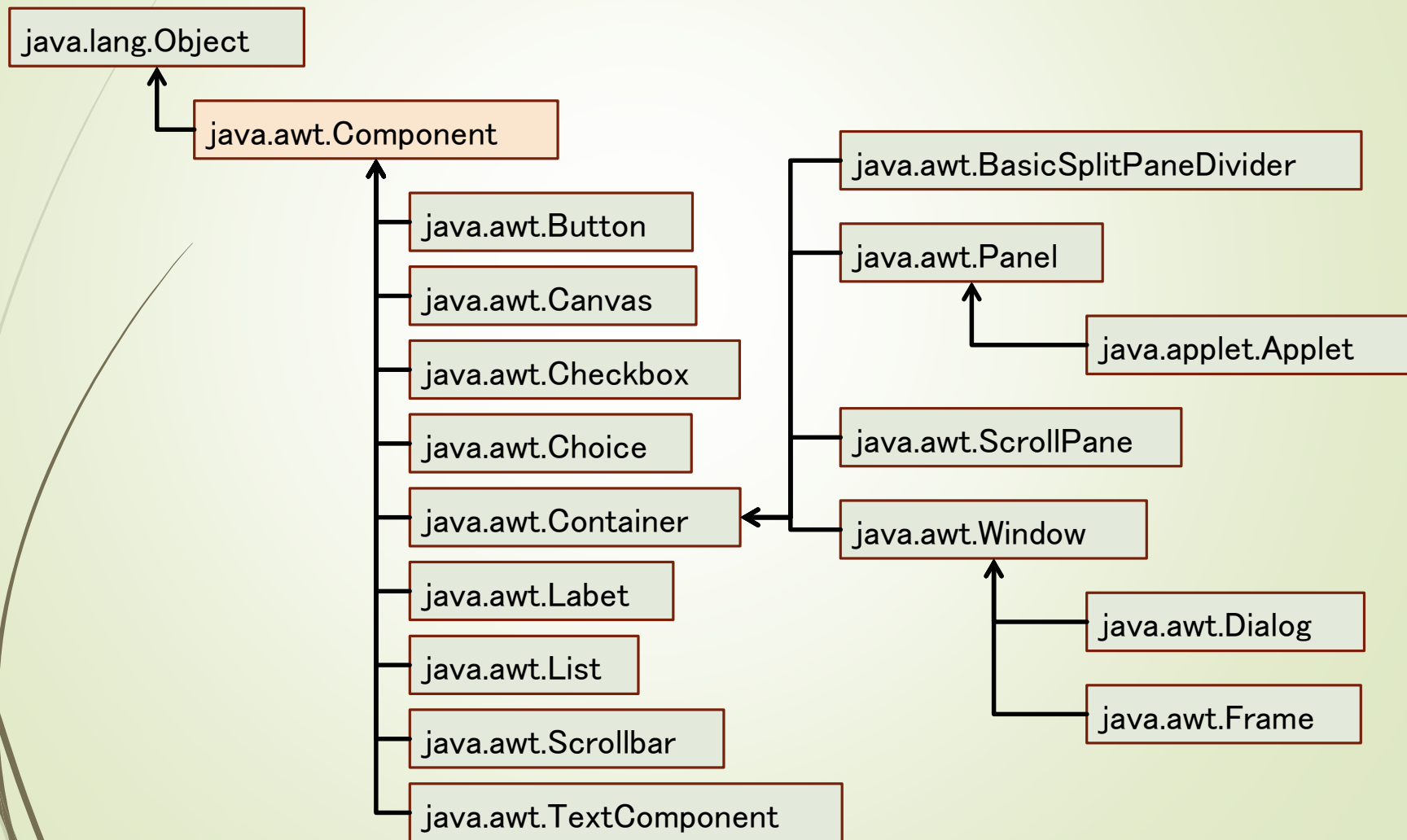
- GUIは様々な部品(widget)で構成
- 部品毎に属性と操作
 - 属性：色、大きさ、 etc.
 - 操作：動作、属性変更、表示、 etc.
- 基本部品はライブラリ化
 - 拡張して使用

java.awt

Abstract Window Toolkit

- 基本グラフィックス
 - 色(Color)、線の属性(BasicStroke)、フォント
- 基本widget
 - パネル、ボタンなど部品群
- 基本イベント(java.awt.event)
 - マウス、キーボード、widgetの属性変化

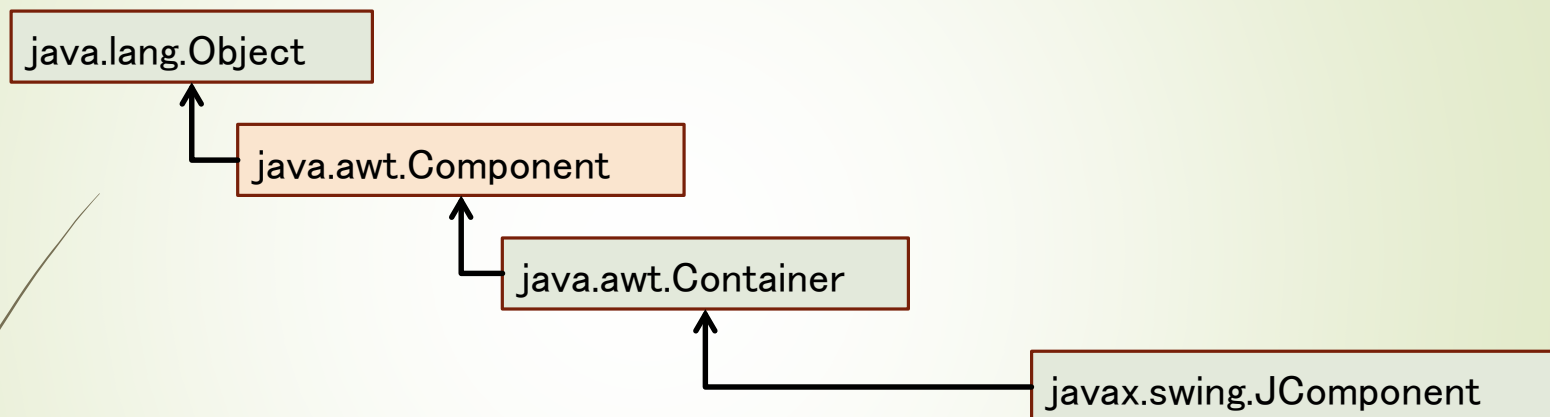
java.awtのwidgets階層



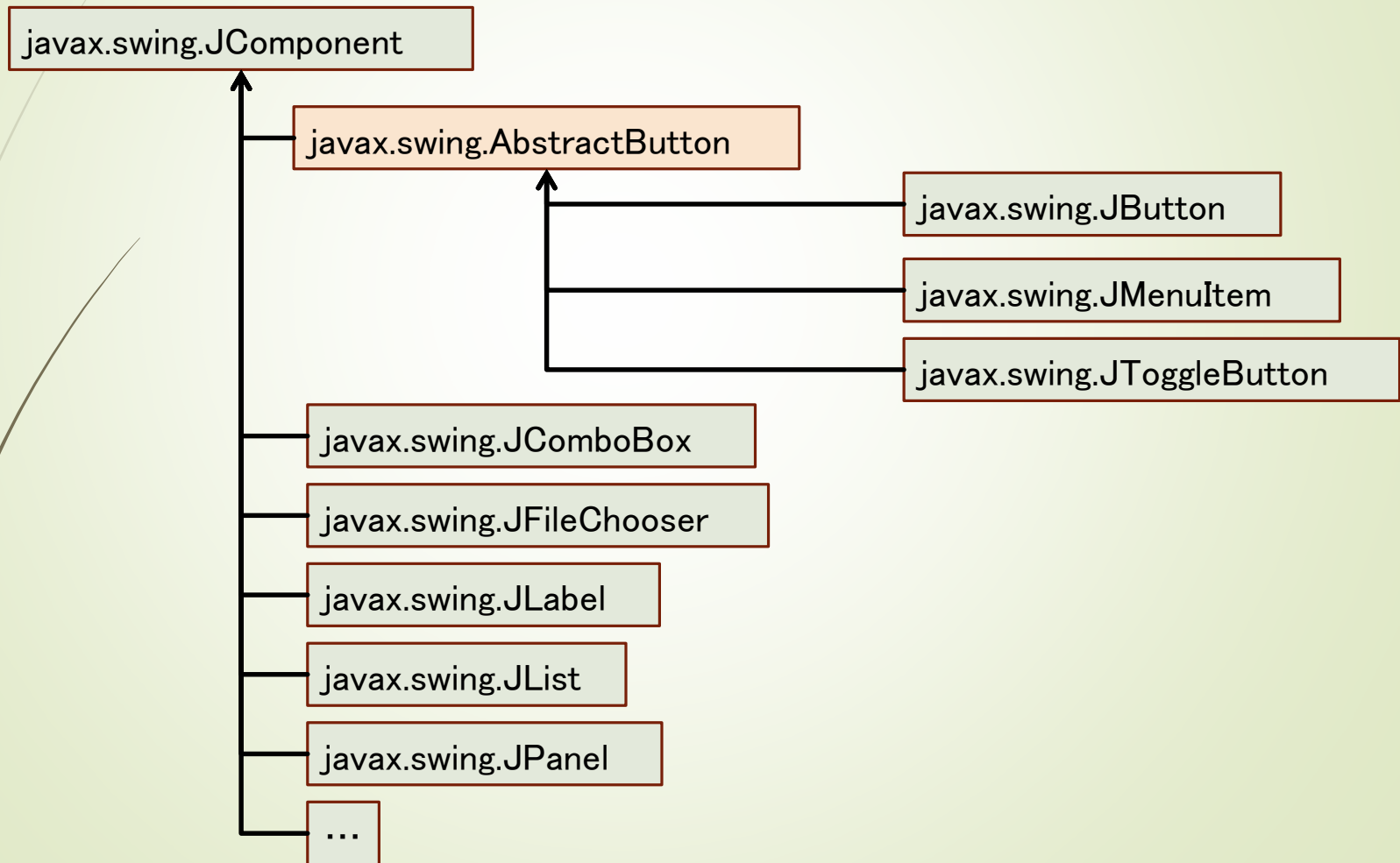
java.awtからjavax.swingへ

- 部品の充実
- プラットフォームからの完全独立
 - ウィンドウマネージャーとの連携
 - Look-and-Feelの分離
- 軽量化
- スレッド対応
 - イベント間通信

javax.swingのクラス階層



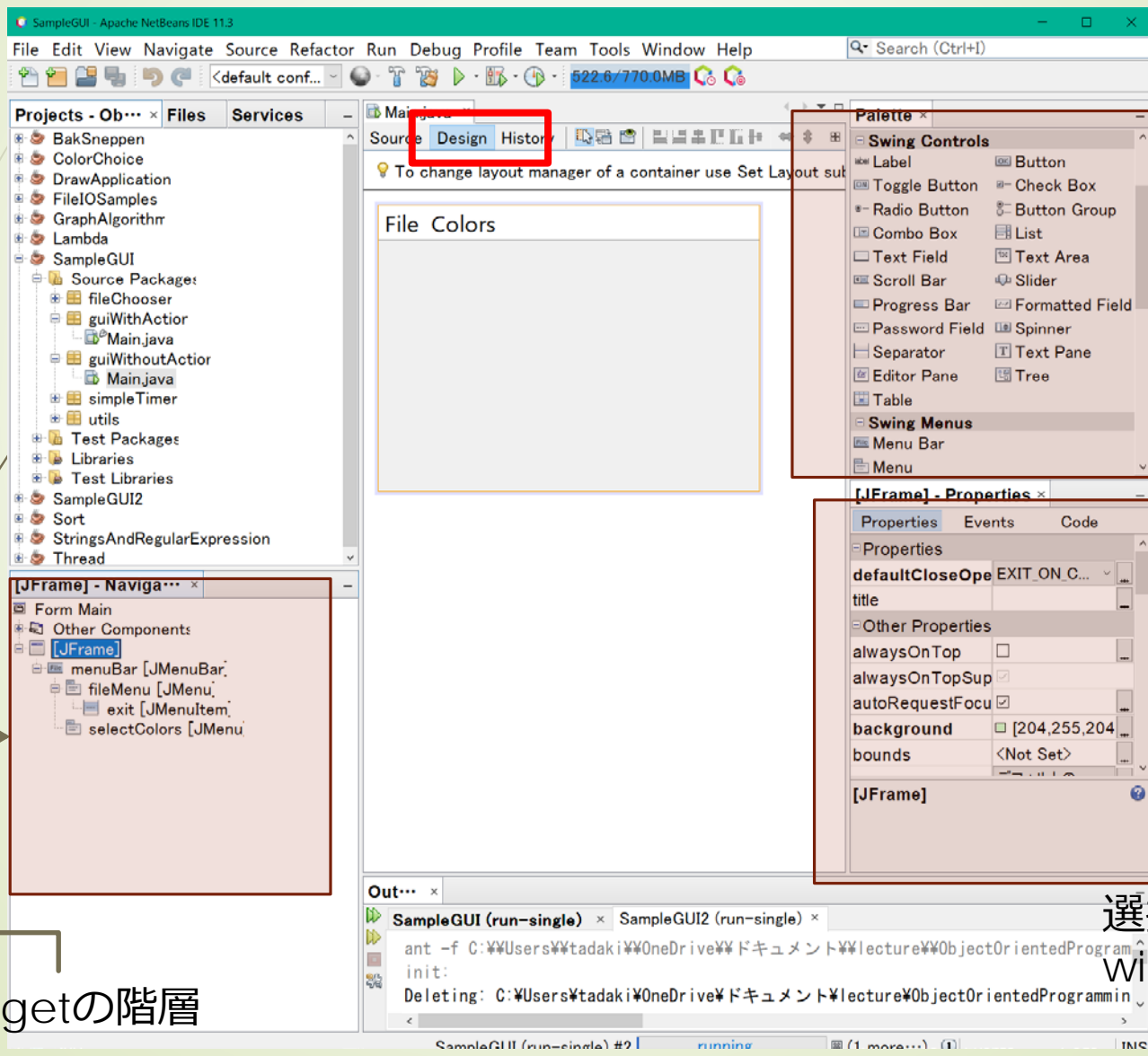
javax.swingのクラス階層



swingコンポーネントの役割

- `javax.swing.JFrame`
 - アプリケーションのメインウィンドウ
 - この中にJPanelを置いて、利用する
- `javax.swing.JMenuBar`
 - アプリケーションの上部のメニューバー
 - `javax.swing.JMenu`を配置
- `javax.swing.JPanel`
 - 様々なwidgetの台（置き場）
 - 図形描画

例：動作の無いGUI：編集画面



widget一覧

widgetの階層

選択した
widgetの属性

NetBeansでGUIを作る

- 通常と同様にプロジェクトを作成する
- JFrame作成
 - 「新規」→「JFrameフォーム」
 - レイアウト設定：「ボーダーレイアウト」
 - JFrameクラスを拡張して利用

widgetの配置

- マウスによる配置
 - 「ナビゲーション」ウィンドウ内で
 - パレットからドラッグ
- JMenuBar作成
 - JMenu追加
 - JMenuItem追加

GUIを作る時の注意

- GUIのプロパティや配置情報はクラス名.formファイルに
 - ソースファイルには編集できない部分
- 部品を拡張した自作クラスも操作可能
- 実際の作成デモンストレーション

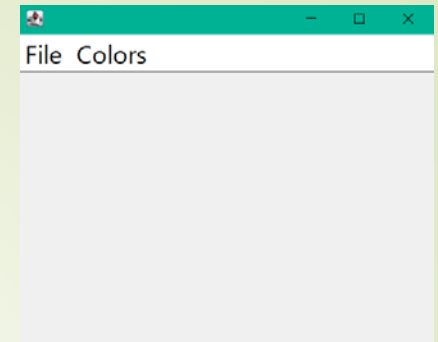
今日のサンプルプログラム

➡ <https://github.com/oop-mc-saga/SampleGUI>

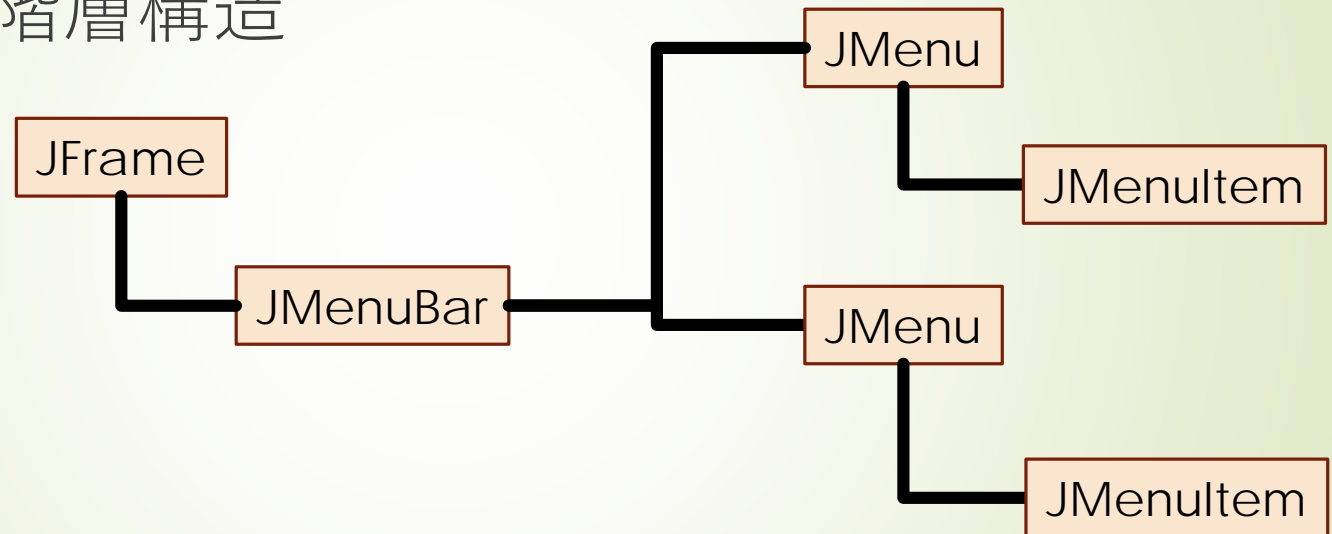
サンプルプログラム

- 動作の無いGUI
 - `guiWithoutAction`
- 動作の有るGUI
 - `guiWithAction`
- ファイルの選択
 - `fileChooser`
- タイマー
 - `simpleTimer`

guiWithoutAction



➡ 階層構造



➡ 配置し、テキストや色を設定


```
11 public class Main extends javax.swing.JFrame {
12
13     public enum Colors { //色を列挙型として定義
14         ORANGE(Color.ORANGE), YELLOW(Color.YELLOW), GREEN(Color.GREEN);
15         private final Color color;
16
17         Colors(Color color) {
18             this.color = color;
19         }
20
21         public Color getColor() {
22             return color;
23         }
24     }
25
26     /**
27      * Creates new form MainWithJMenu
28      */
29     public Main() {
30         initComponents();
31         Font font = new Font("MS UI Gothic", 0, 24);
32         for (Colors m : Colors.values()) {
33             JMenuItem item = new JMenuItem(m.toString());
34             item.setFont(font);
35             selectColors.add(item);
36         }
37     }
}
```

initComponents()は
要素を配置するメソッド。
自動生成されている。

```

36     }
37 }
38
39 /**
40  * This method is called from within the constructor to initialize
41  * WARNING: Do NOT modify this code. The content of this method is alwa
42  * regenerated by the Form Editor.
43  */
44 @SuppressWarnings("unchecked")
45 Generated Code
87
88 /**
89  * @param args the command line arguments
90  */
91 public static void main(String args[]) {
92     /* Set the Nimbus look and feel */
93     Look and feel setting code (optional)
94     //</editor-fold>
114
115
116     /* Create and display the form */
117     java.awt.EventQueue.invokeLater(new Runnable() {
118         public void run() {
119             new Main().setVisible(true);
120         }
121     });
122 }
123

```