

Welcome to

プログラミング演習2

Instructor: Jianhua MA

TA:池邊 曜 and 李 漠 (LI Mo)

- Course HP URL:

<https://cis.k.hosei.ac.jp/~jianhua/course/j2/>

第1回

GUI (Graphics User Interface)

基礎とコンポーネントの作成

キーポイント

- GUIコンポーネント
 - GUIとは何か
 - GUIコンポーネントクラス
 - コンポーネントの作成

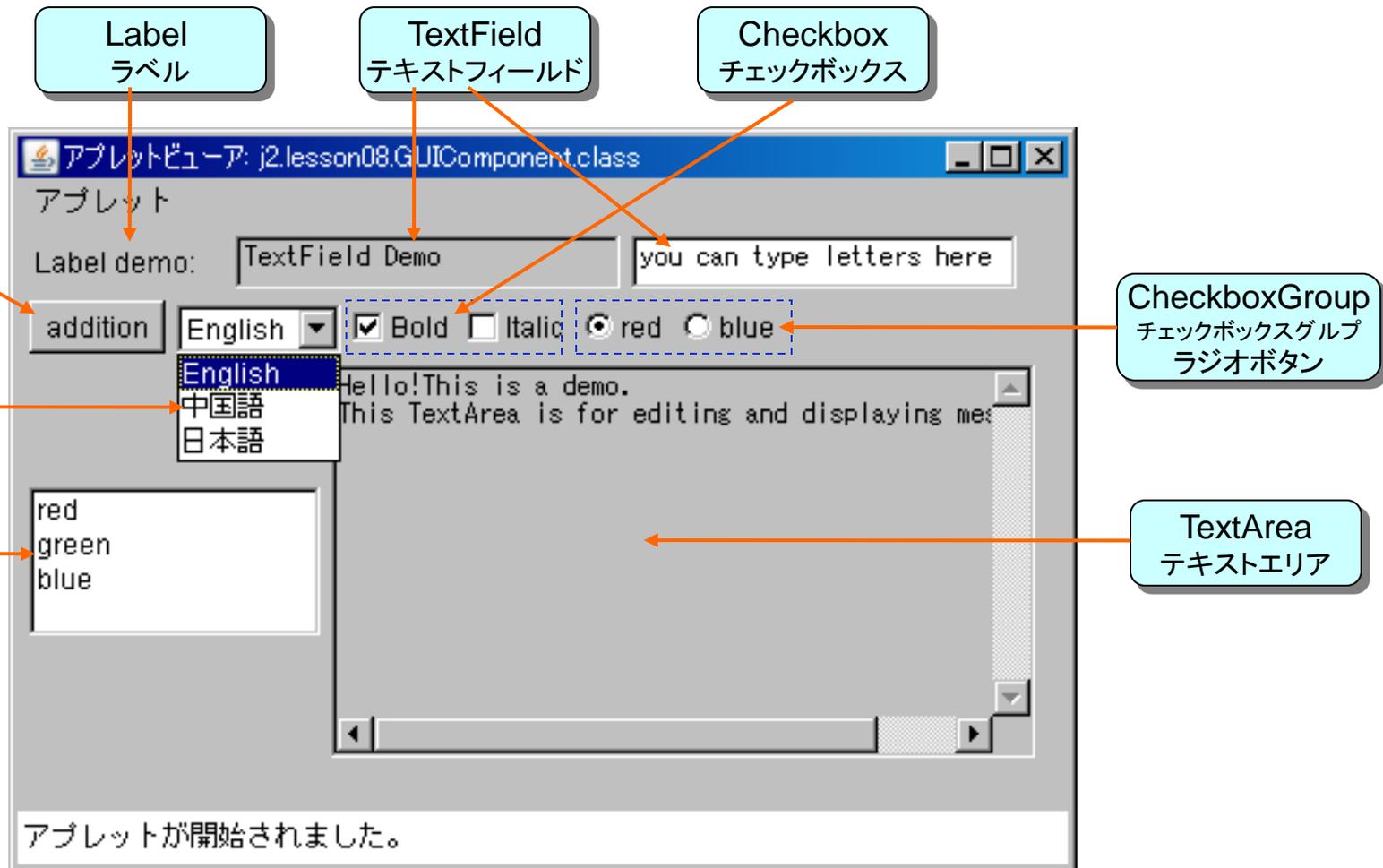
GUI(Graphics User Interface)とは

- Javaではグラフィカルなコンポーネント(部品)を用いたアプリケーション開発用にSwingというGUIライブラリが用意されている
- ボタン、ラベル、テキスト、図形などをグラフィックとして描画する
- AWT(Abstract Window Toolkit)というGUIライブラリも用意されているが、Swingは使用できるコンポーネントなどが大きく増えたもの

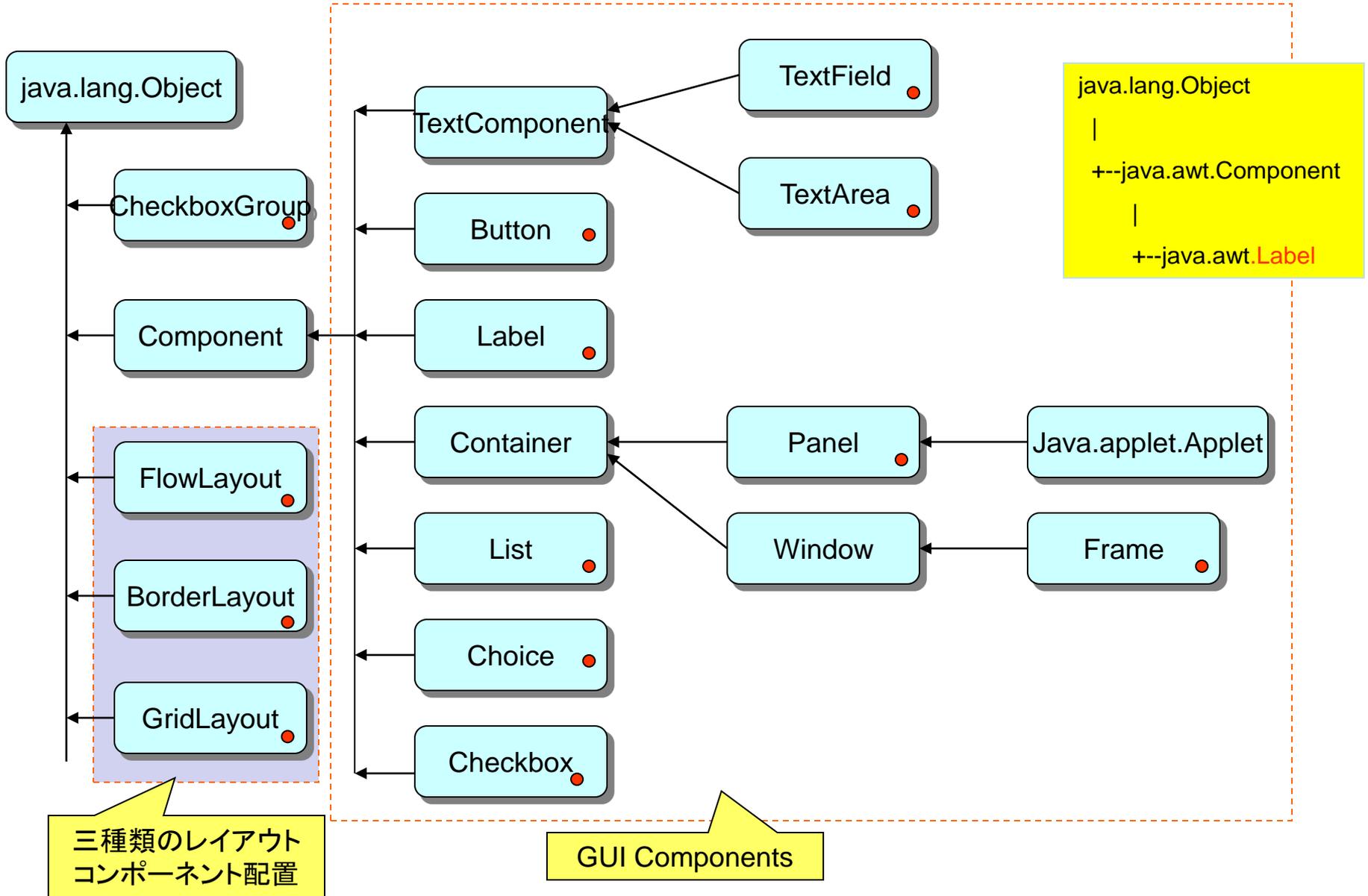


GUI (Graphics User Interface)

GUIコンポーネント(部品)



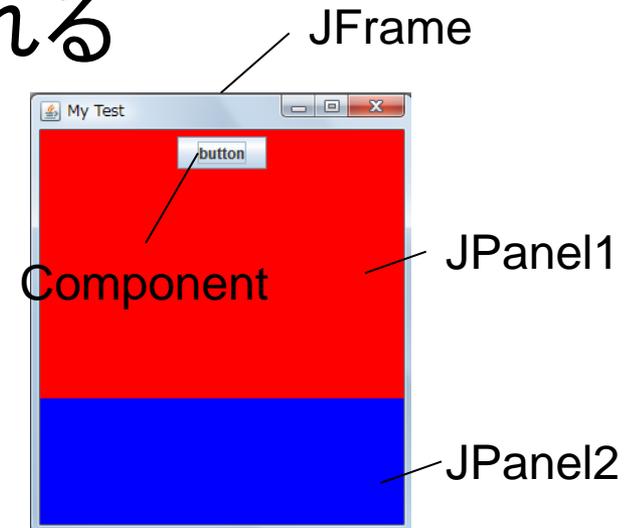
java.awtパッケージ中の GUIコンポーネントとクラス継承階層



GUIコンポーネントの階層

- javax.swingパッケージ中のSwingコンポーネントは以下の3階層に分類される

- トップレベルコンテナ (JFrame等)
 - メインウィンドウとなるコンポーネント、土台の役割
- 中間コンテナ (JPanel等)
 - コンポーネント配置のためのコンテナの役割
- コントロール (JLabel、JButton等)
 - ボタンやラベルなどのGUIコンポーネント



- Swingではこれらの階層のコンポーネントを組み合わせて作成する

GUIコンポーネント

- Swingは以下のようなGUIコンポーネントを持つ
 - JDialog(標準ダイアログ)
 - JButton / JToggleButton(ボタン)
 - JCheckBox / JRadioButton (チェックボックス・ラジオボタン)
 - JMenuItem / JMenuBar / JMenu /
JCheckBoxMenuItem / JRadioButtonMenuItem
(メニュー関連のコンポーネント)
 - JComboBox(ドロップダウンリスト)
 - JLabel / JTextArea / JTextPane(テキスト・イメージ関連)
 - JScrollBar(スクロールバー)
 - JToolBar(ツールバー) etc..

フレームの作成 (JFrameクラス)

- ボタンやラベルなどのコンポーネントを配置するためのフレームを作成する

演習課題を作成し、各要素の機能を理解して下さい

```
import javax.swing.*; //Swingをインポート
```

```
public class Practice01{  
    JFrame jf;
```

```
    public static void main(String[] args){  
        new Practice01();  
    }
```

```
    public Practice01(){  
        jf = new JFrame("My Test");
```

```
        //フレームのサイズと位置を設定  
        jf.setSize(300,350);
```

```
        jf.setLocation(400,200);
```

```
        //フレームを閉じた時にプログラムを終了
```

```
        jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

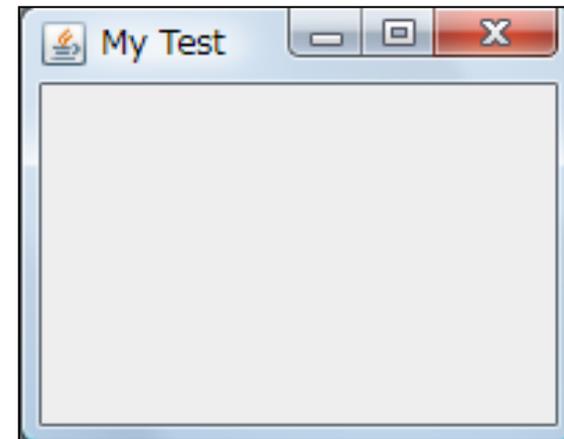
```
        //フレームの表示 (falseで非表示)
```

```
        //コードの最後に記述して下さい(コンポーネントの配置が終わる前に表示してしまうため)
```

```
        jf.setVisible(true);
```

```
    }
```

```
}
```



パネルの作成 (JPanelクラス)

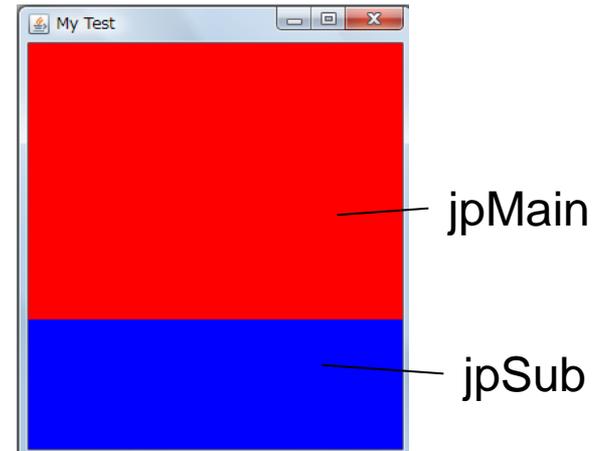
- ボタンやラベルなどのコンポーネントを貼りつけたり、レイアウトを設定することができる

```
public class Practice01{ /* フレームの記述は省略 */
    JFrame jf;
    JPanel jpMain, jpSub;

    public static void main(String[] args){
        new Practice01();
    }
    public Practice01(){
        jpMain = new JPanel();
        //Panelの色を指定
        jpMain.setBackground(Color.RED);

        jpSub = new JPanel();
        //Panelのサイズ Dimensionは幅と高さの2値
        jpSub.setPreferredSize(new Dimension(100,100));
        jpSub.setBackground(Color.BLUE);

        //FrameにPanelを追加と位置の指定
        jf.add(jpMain,"Center");
        jf.add(jpSub,"South");
    }
}
```



色指定はjava.awtの
Colorクラスのオブジェクトで行うので
Importが必要 (import java.awt.Color.*;)

パネルのレイアウト

- デフォルト設定のFlowLayoutや、BorderLayout、GridLayoutなどのレイアウトマネージャが用意されている

```
public class LayoutTest {
    public static void main(String[] args){
        JFrame jf = new JFrame("LayoutTest");
        JPanel jp1, jp2;
        JButton b1,b2,b3,b4,b5,b6,b7,b8;
        //(ButtonとPanelの定義は省略しています)

        jp1.setLayout(new BorderLayout()); //宣言はなくても可
        jp2.setLayout(new FlowLayout());

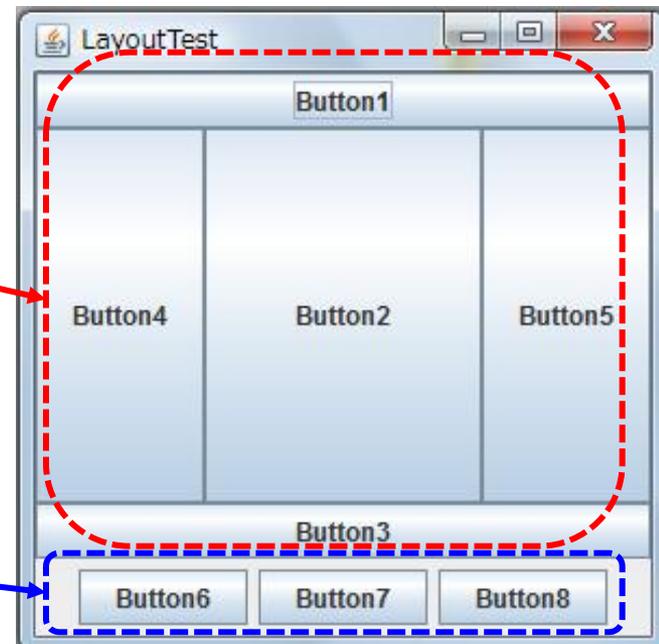
        jp1.add(b1,"North");
        //以下のようにも書ける
        //jp1.add(b1,BorderLayout.NORTH);
        jp1.add(b2,"Center");
        jp1.add(b3,"South");
        jp1.add(b4,"West");
        jp1.add(b5,"East");
        jp2.add(b6);
        jp2.add(b7);
        jp2.add(b8);

        jf.add(jp1,"Center");
        jf.add(jp2,"South");
        jf.setSize(300, 300);
        jf.setVisible(true);
        jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

LayoutTest という名前のクラスを作成し、動作を確認して下さい

jp1

jp2



コンポーネントの作成(ラベル・ボタン・テキストフィールド)

- ボタンやラベルなどのコンポーネントを配置するためのフレームを作成する

```
public class Practice01{
    /* 省略*/
    public Practice01(){
        /* 省略*/
        //jf.add(jpSub,“South”); //(jpSubパネルは使用しない)

        JLabel jl = new JLabel(“This is a label”); //ラベルの作成
        jpMain.add(jl); //Panelに追加

        // ボタンの作成(引数にボタン上に表示されるテキストを与える)
        JButton jb = new JButton(“This is a button in teaching code”);
        jpMain.add(jb);

        //テキストフィールドの作成(引数にテキストフィールド上に表示されるテキストを与える)
        JTextField jtf = new JTextField(“This is a text field in teaching code”, 20);
        jpMain.add(jtf);

        JTextField jtf1 = new JTextField(20);
        jpMain.add(jtf1);
        /* 省略*/
    }
}
```



コンポーネントの前景色・背景色・透明/非透明

- コンポーネントには前景色(文字色)と背景色(コンポーネントの背景)、背景色の透明化の設定ができる

```
public class Practice01{
    /* 省略*/
    public Practice01(){
        /* 省略*/
        JLabel jl = new JLabel(" I'm a label ");
        //文字色の指定
        jl.setForeground(Color.RED);
        //背景の透明・非透明の設定 (falseで透明)
        jl.setOpaque(true);
        jpMain.add(jl);

        JButton jb = new JButton("This is a button in teaching code");
        //文字色と背景色の指定(ボタン以外のコンポーネントでも指定できる)
        jb.setForeground(Color.BLUE);
        jb.setBackground(Color.BLACK);
        jpMain.add(jb);
        /* 省略*/
    }
}
```



I'm a label



This is a button in teaching code

コンポーネントの作成(チェックボックス)

```
public class Practice01{
    /* 省略*/
    public Practice01(){
        /* 省略*/
        JLabel jlCareer = new JLabel(" Please select the careers you like: ");
        jpMain.add(jlCareer);

        JCheckBox[] jcbs = new JCheckBox[2];
        jcbs[0] = new JCheckBox("Teacher");
        jcbs[1] = new JCheckBox("Engieneer");
        for(int i = 0; i < 2; i++){
            //JCheckBoxオブジェクトをそれぞれ追加
            jpMain.add(jcbs[i]);
        }
        /* 省略*/
    }
}
```

Please select the careers you like:

Teacher

Engieneer

コンポーネントの作成(ラジオボックス・コンボボックス)

```
public class Practice01{
    /* 省略*/
    public Practice01(){
        /* 省略*/
        //ラジオボタンのボタングループを作成
        ButtonGroup bg_sex = new ButtonGroup();

        //ラジオボタンのオブジェクトを作成
        JRadioButton jrb_student = new JRadioButton("student");
        //チェックを入れたい場合以下のように指定
        jrb_student.setSelected(true);
        //ラジオボタンのオブジェクトをaddする
        jpMain.add(jrb_student );

        JRadioButton jrb_teacher = new JRadioButton("teacher");
        jpMain.add(jrb_teacher);

        bg_sex.add(jrb_student);
        bg_sex.add(jrb_teacher);

        //コンボボックスを作成
        JComboBox jcb = new JComboBox();
        for(int i = 1; i <= 3; i++){
            //コンボボックスのアイテムを追加
            jcb.addItem(i);
        }
        jpMain.add(jcb);
        /* 省略*/
    }
}
```



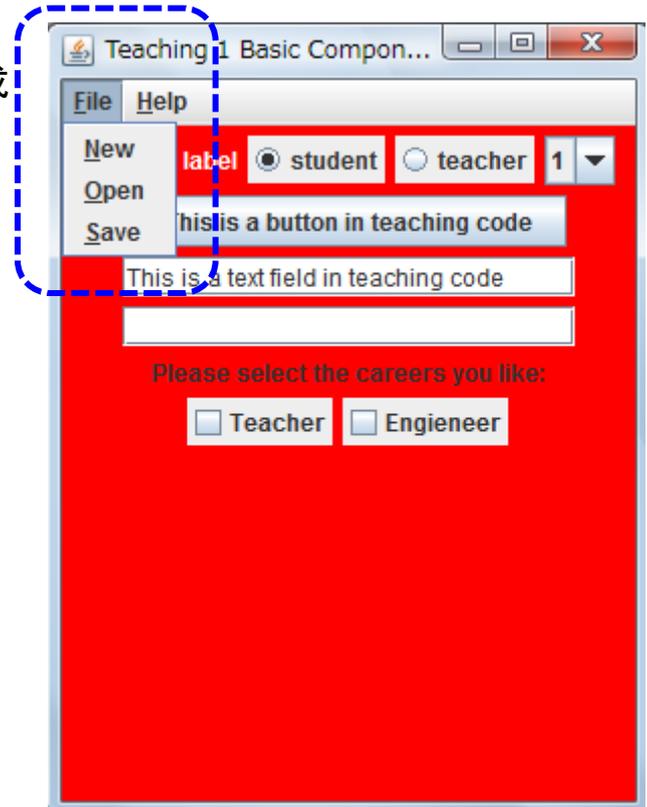
メニューバーの作成(バー、ヘルプ)

- メニューバーの作成と内部のアイテムの作成を行う

```
public class Practice01{
    /* 省略*/
    public Practice01(){
        /* 省略*/
        JMenuBar jmb = new JMenuBar(); //メニューバーの作成

        JMenu menu_file = new JMenu("File"); //メニューの作成
        menu_file.setMnemonic('F'); //(ショートカットキー)
        jmb.add(menu_file); //メニューバーにメニューを追加
        JMenu menu_help = new JMenu("Help");
        jmb.add(menu_help);
        //メニューアイテムの作成・追加
        JMenuItem jmi_new = new JMenuItem("New");
        menu_file.add(jmi_new);
        JMenuItem jmi_open = new JMenuItem("Open");
        menu_file.add(jmi_open);
        JMenuItem jmi_save = new JMenuItem("Save");
        menu_file.add(jmi_save);

        jf.setJMenuBar(jmb); //メニューバーの追加
        /* 省略*/
    }
}
```



ツールバーの作成

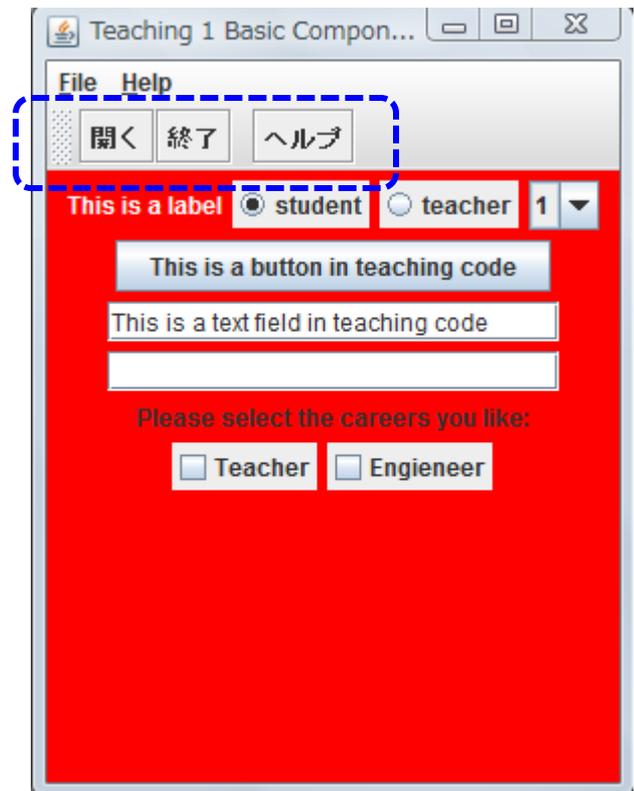
- ツールバーの作成を行う
ツールバーにはボタンや画像などを配置できる

```
public class Practice01{
    /* 省略*/
    public Practice01(){
        /* 省略*/
        JToolBar jtlbar = new JToolBar();

        //ツールバー内のアイテムの作成
        JButton jbtnOpen  = new JButton("開く");
        JButton jbtnExit  = new JButton("終了");
        JButton jbtnHelp  = new JButton("ヘルプ");

        //ツールバーに追加する
        jtlbar.add( jbtnOpen );
        jtlbar.add( jbtnExit );
        jtlbar.addSeparator(); //セパレータの追加
        jtlbar.add( jbtnHelp );

        jf.add(jtlbar,"North"); //ツールバーの配置
        /* 省略*/
    }
}
```



画像の利用(ラベル、ボタン)

- ラベルやボタンには画像を利用することができる

```
public static void main(String[] args){
    JFrame jf = new JFrame("ImageTest");
    JPanel jp = new JPanel(new FlowLayout());

    ImageIcon icon1 = new ImageIcon("new.gif");
    JLabel label = new JLabel(icon1);

    ImageIcon icon2 = new ImageIcon("save.gif");
    JButton button1 = new JButton(icon2);

    ImageIcon icon3 = new ImageIcon("open.gif");
    JButton button2 = new JButton(icon3);

    jp.add(label);
    jp.add(button1);
    jp.add(button2);

    jf.add(jp);
    jf.setSize(300, 100);
    jf.setVisible(true);
    jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
```

ImageTest という名前のクラスを作成し、動作を確認して下さい



JFrameクラスを継承

- GUIの土台となるJFrameをクラスに継承することができる
- JFrameのサブクラスを作成した上で、JFrameのメソッドを拡張したり、新規のメソッドを追加していくことができる

```
import javax.swing.JFrame;

class Practice01_2 extends JFrame{
    public static void main(String args[]){
        //JFrameを継承したサブクラスのオブジェクトを作成
        Practice01_2 jf = new Practice01_2("MyTest2");
    }

    //コンストラクタ内に記述
    Practice01_2(String title){

        JButton btn1 = new JButton("Button1");
        btn1.addActionListener(this);

        //JFrameを継承した場合、以下のように記述することができる
        add(btn1);

        setTitle(title);
        setSize(300,350);
        setLocation(400,200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
}
```

JFrame継承クラスを用いて画面を作成

- 先ほど作成したPractice01を利用して、JFrame継承クラスで再作成する
 - JFrameクラスの使い方だけ変更する

```
import javax.swing.JFrame;

//JFrame jf;    //<-JFrameオブジェクトは必要ない
... ..

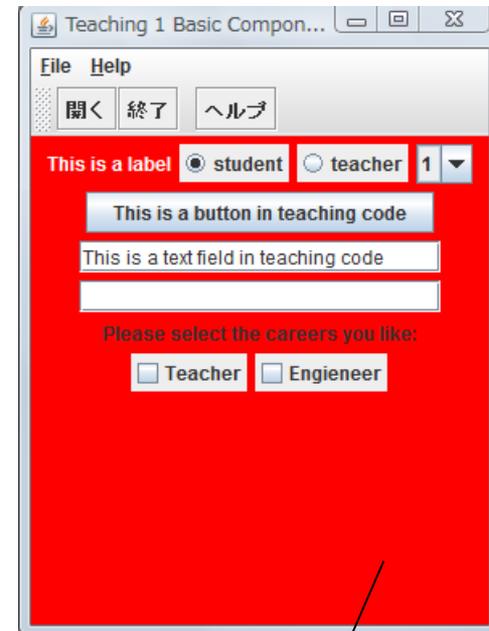
class Practice01_3 extends JFrame{
    public static void main(String args[]){
        new Practice01_3("Practice01_3");
    }
    Practice01_3(String title){
        //タイトルの設定
        setTitle(title);

        ....
        ....

        //JFrameを継承しているので、"jf.add~"や"jf.set~"
        //というようにする必要がない
        setJMenuBar(jmb);

        add(jpMain, "Center");

        setSize(300,400);
        setLocation(400,200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
}
```

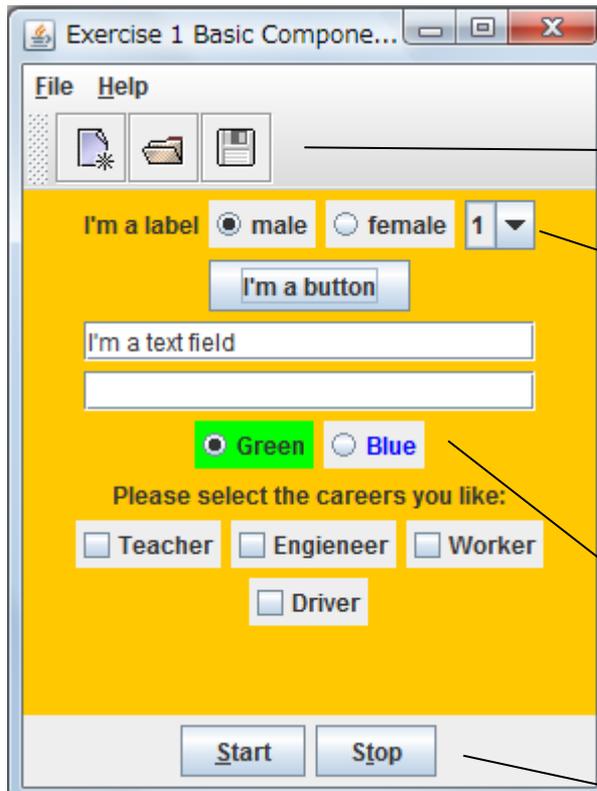


同じ画面を作成できる
JFrameオブジェクトを作成する一手間
がないので、コードを綺麗に記述できる

JFrameを継承する方法も、JFrameオ
ブジェクトを作成する方法も、実際に
やっていることに大差はない

Exercise 1

- 次のようなデザインのGUIを「Ex1BasicComponent」という名前のクラスで作成して下さい(JFrameのサイズ300x400)



メニューバーを作成し、アイテムを追加する

ツールバーを作成し、画像を用いたアイコンを挿入する

コンボボックスのアイテムは「1~9」の数字を追加する

ラベル、ボタン、ラジオボタン、コンボボックス、チェックボックス、テキストフィールドを配置する

背景色・テキストの色を変更したコンボボックスを作成する

上のパネルと別のパネルを用意しボタンを配置する
また、図にあるようなレイアウトにする

