

第五回 演習問題 (GUI1)

諸注意

- 今回、ファイルの提出は行わない。
- 課題ができた段階で教員もしくは TA を呼びその場でチェックを受ける。

課題 1

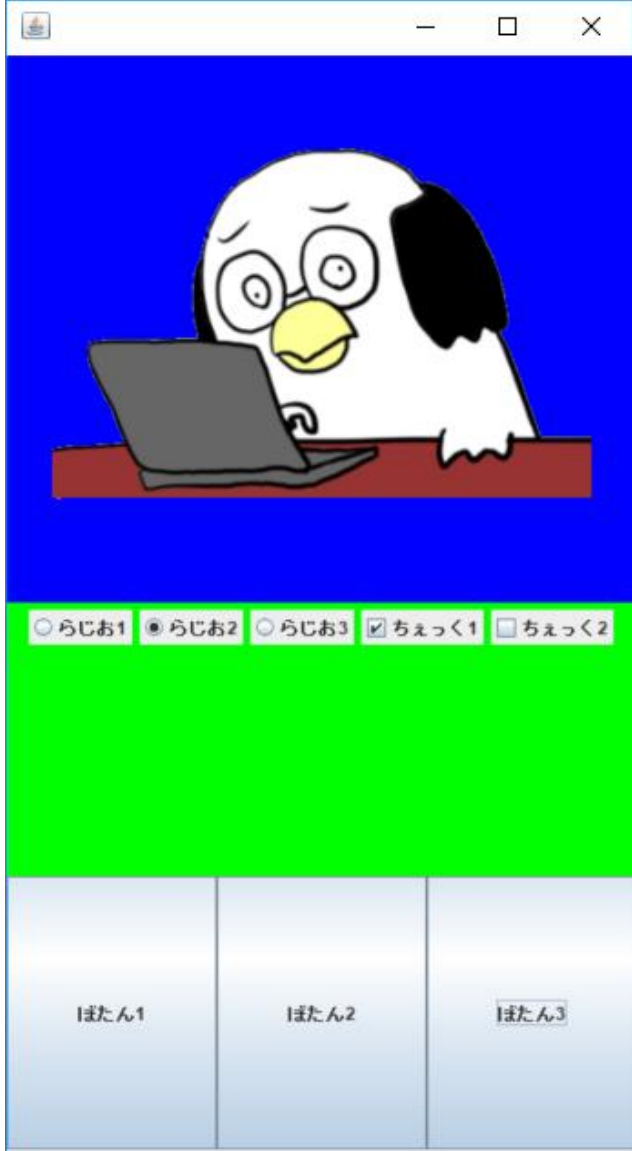
	動作画面例	問題設定
1		<p>図 1 のようなフレームを次の手順で作成する。</p> <ul style="list-style-type: none"> ● フレームのサイズを横 450*縦 800 とする。 ● コンテナの上下に配置できるように, GridLayout を設定する。 ● 上半分を扱う JPanel (header) と, 下半分を扱う JPanel (footer) をコンテナに追加する。 ● header を青背景にする。 ● header に JLabel を用いて画像を中央に表示する。画像は演習問題とともに配布する。 ● GridLayout を用いて, footer に更に上下に配置できるようにする。 ● footer 上部は緑背景の JPanel (upper) を配置し, 図 1 の通りのコンポーネントを配置する。 ● footer 下部には JPanel (lower), ボタン 1~3 は均等の幅に配置する。
	<p>上記設定を満たすフレームとして KadaiFrame.java を開発せよ。なお継承や実装は必要に応じて実施すること。</p>	
	<p>ヒント</p> <p>以下のようなコンテナ, パネルの入れ子構造になっている。</p> <pre> コンテナ(Container) -- header(JPanel) -- img(JLabel) -- footer(JPanel) -- upper(JPanel) -- radio...等 -- lower(JPanel) -- button*3 </pre>	

図 1. 動作画面例

課題 2

2	問題設定	<p>課題 1 で開発した KadaiFrame に対して次の機能を拡張せよ。なお、継承や実装は必要に応じて実施すること。</p> <p>ボタン 1 を押すと青背景のパネルに表示されている画像が 違う画像に切り替わる。ボタン 2 を押すとさらに違う画像に切 り替わる。ボタン 3 を押すとボタン 1, ボタン 2 が押せなくな る（もう一度押すとボタン 1, 2 が押せるように戻る）。</p>
---	------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

課題 1, 2 : 解答例

```

public class KadaiFrame1 extends JFrame implements ActionListener, ItemListener{
    private JButton button1, button2, button3;
    private JPanel header, footer, upper, lower;
    private JLabel imageLabel;
    private ImageIcon image1, image2;

    public KadaiFrame1(){
        // フレームの設定関連
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(450, 800);
        this.getContentPane().setLayout(new GridLayout(2, 1));
        this.setVisible(true);

        // 上部分の パネルを作成しコンテナに追加
        this.header = new JPanel();
        this.getContentPane().add(this.header);

        // 背景色を青にし、画像のJLabelをheaderに登録
        this.header.setBackground(Color.BLUE);
        this.imageLabel = new JLabel(new ImageIcon("25.png"));
        this.header.add(imageLabel);

        // 下部分の パネルを作成
        this.footer = new JPanel();
        this.footer.setLayout(new GridLayout(2, 1));
        this.getContentPane().add(this.footer);

        // 上下部分の パネルを作成し背景色を緑にしてfooterに登録
        this.upper = new JPanel();
        this.upper.setLayout(new FlowLayout());
        this.upper.setBackground(Color.GREEN);
        this.footer.add(this.upper);

        // ラジオボタンをグループ化し、upperに登録
        JRadioButton radio1 = new JRadioButton("らじお1");
        JRadioButton radio2 = new JRadioButton("らじお2", true);
        JRadioButton radio3 = new JRadioButton("らじお3");
        ButtonGroup bg = new ButtonGroup();
        bg.add(radio1); bg.add(radio2); bg.add(radio3);
        this.upper.add(radio1); this.upper.add(radio2);
        this.upper.add(radio3);

        // チェックボックスをupperに登録
        JCheckBox check1 = new JCheckBox("ちえっく1", true);
        JCheckBox check2 = new JCheckBox("ちえっく2", false);
        this.upper.add(check1); this.upper.add(check2);
    }

```

```

// 最下部用のパネルを作成しfooterに登録
this.lower = new JPanel();
this.lower.setLayout(new GridLayout(1, 3));
this.footer.add(this.lower);

// ボタンを作成し, ActionListener登録後 lowerに登録
this.button1 = new JButton("ボタン1");
this.button2 = new JButton("ボタン2");
this.button3 = new JButton("ボタン3");
this.button1.addActionListener(this);
this.button2.addActionListener(this);
this.button3.addActionListener(this);
this.lower.add(button1);
this.lower.add(button2);
this.lower.add(button3);

// 以下課題2用
String[] array = {"ジャンプ", "ねこ", "カレー", "ラーメン", "車"};
JComboBox combo = new JComboBox(array);
this.upper.add(combo);
combo.addItemListener(this);

// 表示画像を事前に準備しておく (毎回インスタンス化しなくて済む)
this.image1 = new ImageIcon("02.png");
this.image2 = new ImageIcon("05.png");
}
@Override
public void actionPerformed(ActionEvent e) {
    if(e.getSource() == this.button1){
        // setIcon()メソッドを用いて画像を変更する.
        // imageLabel = new JLabel(new ImageIcon("xx.png"))と
        // 書くとheaderにaddしてはimageLabelと今インスタンス化
        // したJLabelが紐付かなくなってしまうので注意が必要である.
        this.imageLabel.setIcon(this.image1);
    }else if(e.getSource() == this.button2){
        this.imageLabel.setIcon(this.image2);
    }else if(e.getSource() == this.button3){
        // トグルの実現にはisEnabled()を反転させるだけで良い
        this.button1.setEnabled(!this.button1.isEnabled());
        this.button2.setEnabled(!this.button2.isEnabled());
    }
}

// 以下課題3用
@Override
public void itemStateChanged(ItemEvent e) {
    // e.getStateChange()でどんなイベントが起こったのかを検出できる
    // ItemEvent.SELECTEDとすることで, 選択された時というイベントのみ
    // 検出し, 選択が解除された場合を除外している.
    if(e.getStateChange() == ItemEvent.SELECTED){
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                // MyFrameのインスタンスを生成する
                // コンストラクタの引数で文字列を送る
                // コンボボックスで選んだ直後はe.getItem()で取得する.
                Alert alert = new Alert((String)e.getItem());
            }
        });
    }
}
}
}

```

課題 3

3	問題設定	<p>課題2を拡張し、ポップアップのような機能を実現する。まず、新しく Alert.java を開発する。Alert は横 200*縦 200 フレームであり、一つラベルが置かれている。</p> <p>課題2で開発した KadaiFrame の緑背景のパネルに、新たにコンボボックスを追加する。コンボボックスの選択肢は、各自の好きなものを5個羅列することとする（食べ物でもキャラクターでも何でもよい）。コンボボックスの選択された項目が変更されると、Alert のフレームが新たに表示され、Alert 内のラベルの文字列を、コンボボックスで選択した文字列に変更して表示する。</p> <p>例えば、コンボボックスで「週刊少年ジャンプ」を選択したとすると、新たにフレームが表示され、中央のラベルには「週刊少年ジャンプ」と表示されている。</p>
	ヒント	<p>講義内では紹介していないメソッドを使うシーンがいくつかあるので、ググって対応すること。躓きが予想される個所と原因をいくつか紹介しておく。</p> <ul style="list-style-type: none"> ● コンボボックスで選択された項目が取得できない（エラーがでる）。→e.getSource()ではない。型にも注意。 ● Alert のフレームが二つ出る。→itemStateChanged()は項目を選択されたときだけでなく、選択が解除された時にも発生する。引数の ItemEvent のメソッドを使って、選択された時だけを抽出するとよい。 ● Alert の×を押すと全部消えてしまう。→×を押したときの動作の設定をどこかでやったはず。

課題3：解答例

```

public class Alert extends JFrame{
    private JLabel label;
    public Alert(String text){
        // フレームの設定関連
        // ×を押した時にアプリを終了せずフレームだけ閉じる
        this.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
        this.setSize(200, 200);
        this.getContentPane().setLayout(new FlowLayout());
        this.label = new JLabel(text);
        this.getContentPane().add(this.label, BorderLayout.CENTER);
        this.setVisible(true);
    }
}

```