



Swingによるデスクトップアプリケーション開発（BMI診断）-JavaSE1.8



office・M

2024年9月1日 19:59



Java8のSwing環境でデスクトップアプリケーションの開発方法を学ぶ講座をシリーズで提供しています。今回はWindowBuilder（Swingデザイナー）を使ってBMI診断アプリケーションを作成します。BMI診断とは身長と体重からボディマス指数（BMI）を算出して体型（肥満度）を診断するものです。

2024年9月よりECLIPSEのバージョンを最新版（Version: 2024-06 (4.32.0)）に変更しました。

▼ 目次

外部設計

内部設計

処理ロジック

実装準備

プロジェクトの作成

実装

ひな形の作成

GUI実装

イベント実装

実装変更

単独起動

実行可能JARファイル

最後に

外部設計

WindowBuilderのSwingデザイナーで図1のようなGUIを作成します。



図1. HealthCheckアプリケーションのGUI画面

内部設計

処理ロジック

BMI診断ボタンをクリックすることでアクションイベントが発生し、BMIを算出する処理ロジックを実装したメソッドが起動する。

BMI処理ロジック

1.BMI値を計算する。

BMI=体重(Kg)÷（身長(m)×身長(m)）※新著の単位はcmではなくmである

2.算出したBMI値から次の基準で体形（肥満度）を判定する。

BMI値が18.5未満の場合は体型は「やせ型」

BMI値が18.5以上で25未満の場合は体型は「普通」

BMI値が25以上の場合は体型は「肥満」

実装準備

プロジェクトの作成

Eclipseのメニューバーより

ファイル→新規→Javaプロジェクト→「SwingHealthCheck」プロジェクトを作成→図2の内容で設定する

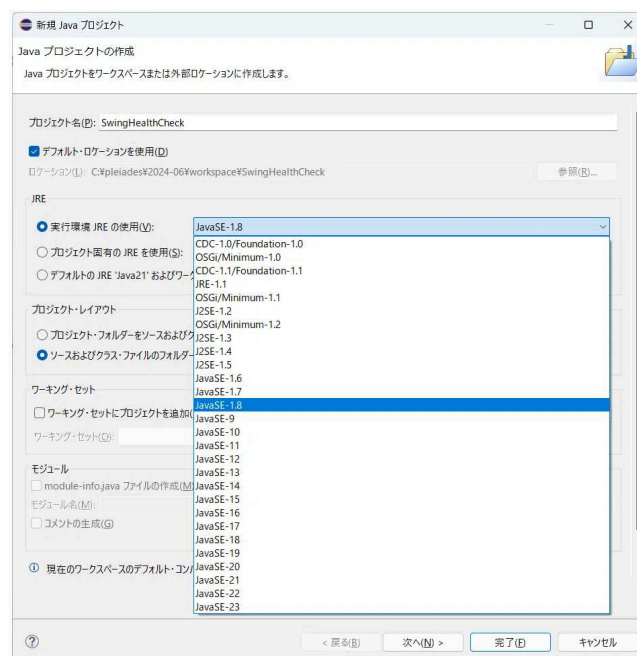


図2. SwingHealthCheckプロジェクト設定

※作成済みであればこの処理は必要ありません。

以下画面のスクリーンショットはライトテーマで取得します。

（ライトテーマの設定方法）

Eclipseのメニューバーより

ウィンドウ → 設定 → 一般 → 外観 → ルック&フィール → ライト
→ 適用して閉じる → Eclipseの再起動がかかります

実装

ひな形の作成

WindowBuilderを用いてSwingアプリケーションのスケルトン（骨格）を自動生成させます。

Eclipseパッケージ・エクスプローラより

SwingHealthCheckプロジェクトを右クリック→新規→その他

→ WindowBuilder → Swingデザイナー → JFrameを選択 → 次へ

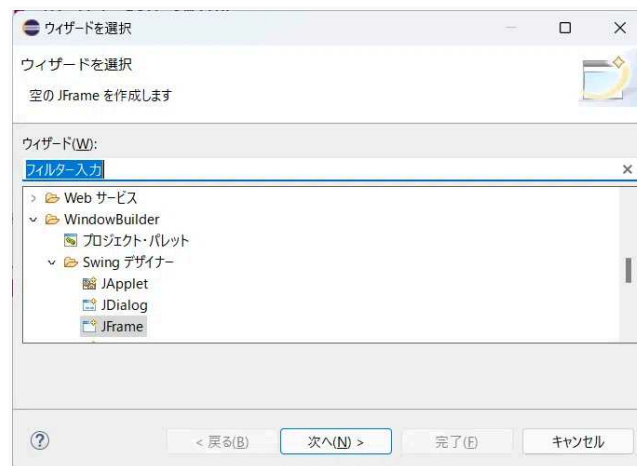


図3. JFrameウィザードの選択

以下の内容で作成

パッケージ : jp.ict.aso.swing

名前 : HealthCheck

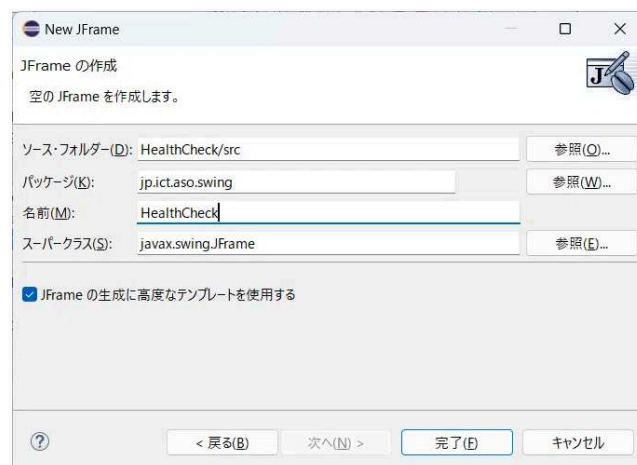


図4. Swingアプリケーションの生成

GUI実装

自動生成されたプログラム（スケルトン）からGUIのデザインを実装します。

パレットと構造（コンポーネント、プロパティ）のViewを利用するのがコツです。デザインイメージは設定反映の参考としてとらえた方が良いでしょう。

①画面中央下部にあるデザインタブでソースコード編集画面からSwingデザイナーに切り替えます。

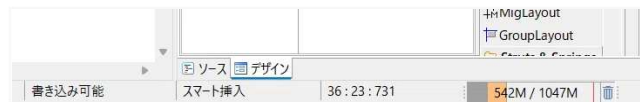


図5. Swingデザイナーに切り替え

②contentPaneのLayoutプロパティをBorderLayoutに設定します。

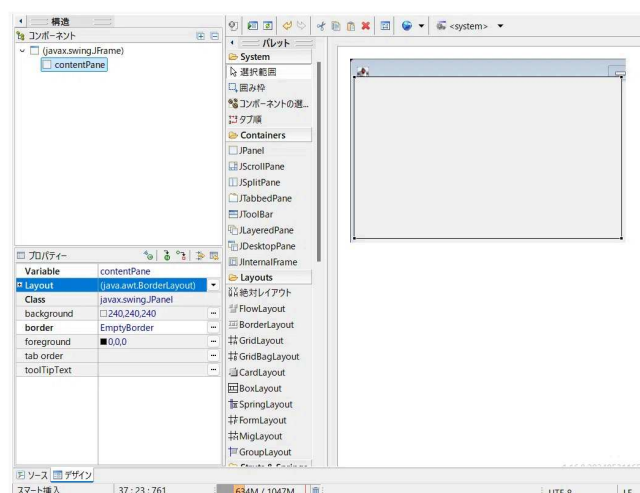


図6. コンテンツペインの設定

③contentPaneの「North」の位置にGUI部品のJPanelをパレットから配置します。JPanelのLayoutプロパティはGridLayoutとします。

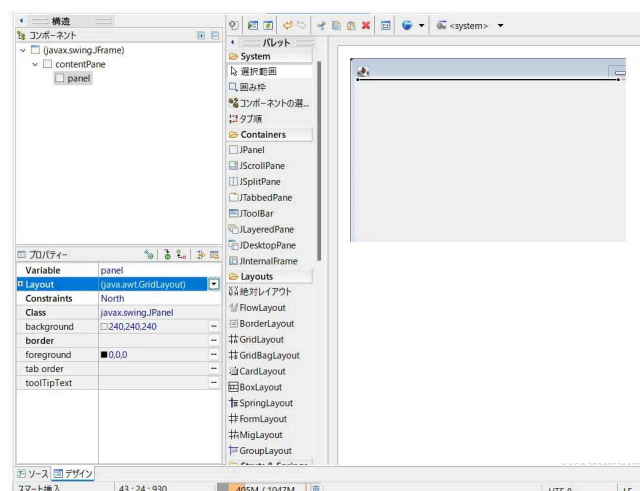


図7. JPanelの設定

④JPanelのLayoutプロパティを + マークから展開しcolumnsを3に、rowsを2に設定します。

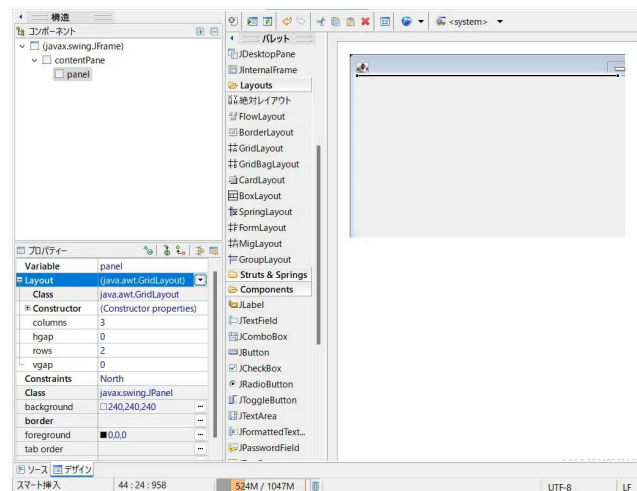


図8. 列と行の設定

⑤JPanelにGUI部品のJLabel、JTextFieldを順番にパレットから配置します。それぞれのtextプロパティを図9のように変更します。

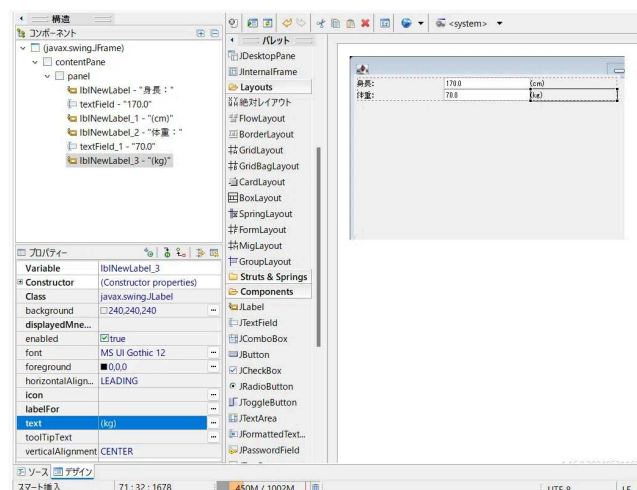


図9. ラベルとテキストフィールドの配置とテキストの設定

⑥contentPaneの「Center」の位置にGUI部品のJLabelをパレットから配置します。あわせてtextプロパティも「BMI診断結果？」に変更します。

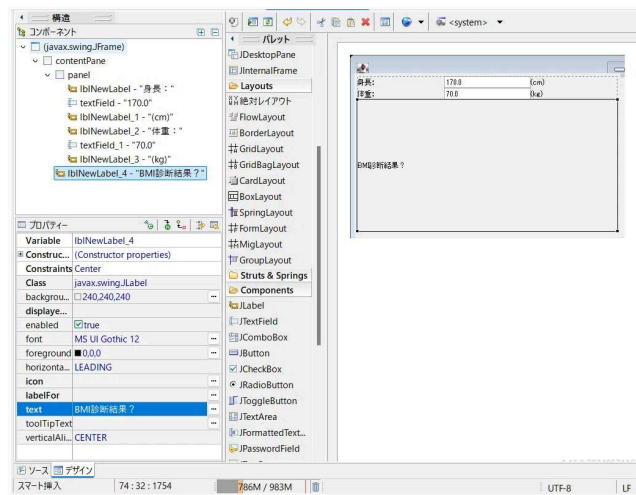


図10.BMI診断結果ラベルの配置

⑦contentPaneの「South」の位置にGUI部品のJButtonをパレットから配置します。あわせてtextプロパティも「BMI診断実行」に変更します。

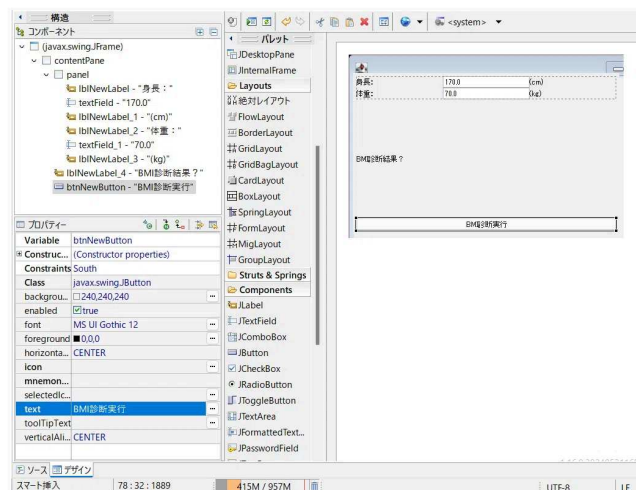


図11. BMI診断実行ボタンの配置

⑩ボタンにイベントリスナーを対応付けます。パレットのSwingActions内にある「新規」のリスナーを選択してボタンをクリックすることで対応付けられます。

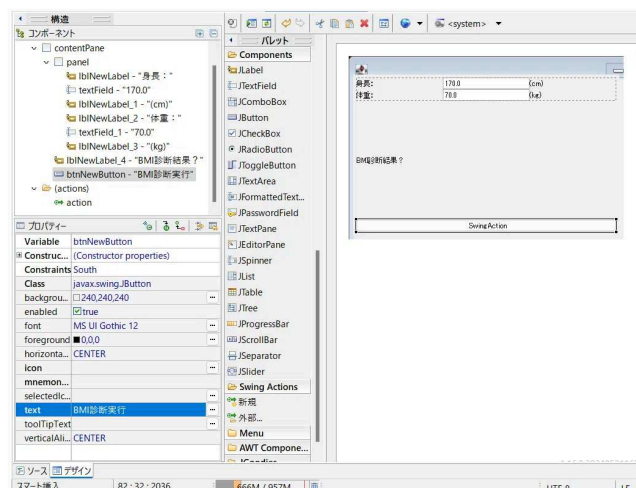


図12. イベントリスナーの対応付け

イベント実装

ソースタブに変更します。

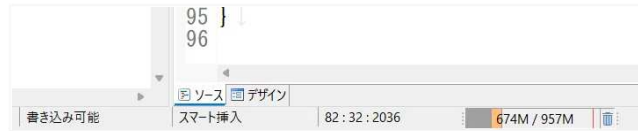


図13. ソースタブに変更

①「BMI診断実行ボタン」のイベントのソース部分を変更します。

```
private class SwingAction extends AbstractAction {  
    public SwingAction() {  
        putValue(NAME, "BMI診断実行");  
        putValue(SHORT_DESCRIPTION, "BMIの計算を実行します");  
    }  
    public void actionPerformed(ActionEvent e) {  
        double height = Double.parseDouble(textField.getText());  
        double weight = Double.parseDouble(textField_1.getText());  
        double bmi=(double) Math.round(weight/(height/100.0*height/100.0)*100)/100;  
        double bmi=weight/(height/100.0*height/100.0);  
        if(bmi<18.5){  
            lblNewLabel_4.setText("BMI値は"+String.valueOf(bmi)+"です。「やせ型」です。");  
        }else if(bmi<25){  
            lblNewLabel_4.setText("BMI値は"+String.valueOf(bmi)+"です。「普通型」です。");  
        }else {  
            lblNewLabel_4.setText("BMI値は"+String.valueOf(bmi)+"です。「肥満型」です。");  
        }  
    }  
}
```

図14. BMI診断実行ボタンのイベント内容

②フィールド変数を変更します。//kokoの部分を追加します。

```
public class HealthCheck extends JFrame {  
    private static final long serialVersionUID = 1L;  
    private JPanel contentPane;  
    private JTextField textField;  
    private JTextField textField_1;  
    private final Action action = new SwingAction();  
    private JLabel lblNewLabel_4; //koko  
    /**  
     * Launch the application.  
     */  
}
```

図15. フィールド変数の変更

③ローカル変数の宣言になっている部分を変更します。//kokoの部分を変更します。

```
JLabel lblNewLabel_3 = new JLabel("(kg)");  
panel.add(lblNewLabel_3);  
lblNewLabel_4 = new JLabel("BMI診断結果?"); //koko  
contentPane.add(lblNewLabel_4, BorderLayout.CENTER);  
JButton btnNewButton = new JButton("BMI診断実行");  
btnNewButton.setAction(action);  
contentPane.add(btnNewButton, BorderLayout.SOUTH);
```

図16. ローカル変数の宣言変更

④実行確認します。エディタの画面内で右クリック → 実行 → Javaアプリケーションで実行されます。



図17. 実行方法

⑤BMI診断実行ボタンのクリックで結果が表示されるか確認します。



図18. BMI診断結果の表示

ここまでの実装ではいろいろと不具合があるようです。
とりあえず「タイトルがない」「画面の大きさが任意に変えられてしまう」の2点を修正します。

実装変更

①フレームにタイトルを追加します。//kokoの部分を追加します。

```
/**
 * Create the frame.
 */
public HealthCheck() {
    setTitle("BMIチェック"); //koko
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 300);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
}
```

図19. タイトルの追加

②画面（フレーム）の大きさを固定します。//kokoの部分を追加します。

```
/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                HealthCheck frame = new HealthCheck();
                frame.setVisible(true);
                //枠固定
                frame.setResizable(false); //koko
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}
```

図20. 画面サイズの固定

③実行して動きを確認します。これで完成しました。



図21. BMI診断アプリケーション完成

単独起動

実行可能JARファイル

①せっかくですので、単独で起動できるアプリケーションにエクスポートしましょう。Java1.8以上のJREの環境がWindowsのPCにインストールされていればダブルクリックで起動できます。

Eclipseパッケージ・エクスプローラより
SwingHealthCheckプロジェクトを右クリック → エクスポート
→ Java → 実行可能JARファイル → 次へ
→ 以下のように設定する → 完了

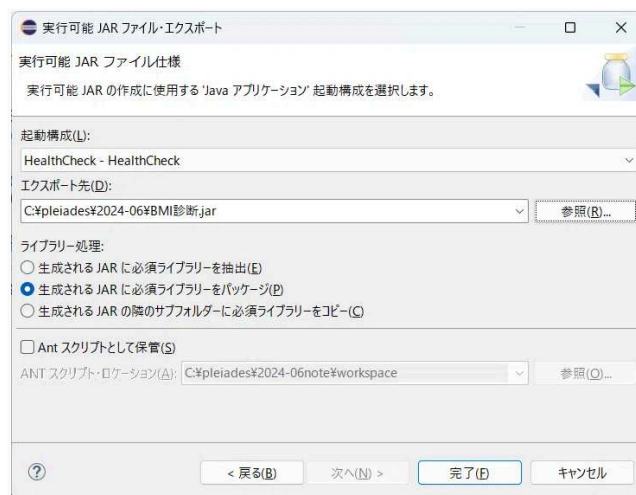


図22. 実行可能JARファイルの設定

以下のような警告が出る場合がありますが、気にしません。



図23. 警告ダイアログ

作成されたjarファイルをダブルクリックするとBMI診断アプリが起動します。



図24. 実行可能JARファイル

最後に

以上でSwingデザイナーを使って身長と体重からBMI値を計算するデスクトップアプリケーションを作成できました。