◆簡単な GUI 作品制作例 電卓 ステップ 1 (設計)
http://manabu.quu.cc/up/jv/eJ1420m0.htm
簡単な作品として、右のような電卓を作成します。

この作品を作る方法は、幾通りもあるでしょう。
しかし、変更などや再利用を考慮すると、作り方は絞られます。

まずこの電卓のフレームで、作品のサイズなどを決定するクラスを作ります。 また、これが作品の main があるクラスです。 このクラスの名前を SimpleCalculator とします。
このフレームで使うパネルに、計算に使う数値入力用のテキストフィールド、 ボタンや計算結果を表示するラベルなどを配置します。 そして、そのボタン用のインターフェイスを implements します。

さてこのパネルのプログラムは、ボタンなどの部品の配置と、 各ボタンのクリックインターフェイスの実装に分かれます。そしてこのインターフェイスも機能は、計算対象の数値入力用のボタンと、 演算用のボタン用に分けることができます。クラスの作り方は、色々と考えられるでしょうが、ここではこれらプログラムの 処理内容により分けて、後々の変更や再利用しやすいようにクラスを検討します。それにより、『ボタンの配置と、数値入力用ボタンのインターフェイス実装』用クラスを SimpleCalcPanel の名前とし、 このクラスを派生させて、『演算関連ボタンのインターフェイスを実装』用 クラスを NomalCalcPanel の名前で作ります。

以上で決めた作品のクラス図を以下に示します。

クラス名	機能
SimpleCalculator	概観用のフレームで、NomalCalcPanel オブジェクトを配置する
SimpleCalcPanel	ボタンを配置して、数値入力用ボタンの ActionListener を実装
NomalCalcPanel	SimpleCalcPanel のサブクラスで、演算用ボタンの ActionListener を実装

作品用に、work のパッケージを使います。
Java 言語の初心者であれば、次のような初期の実験プログラムから作成して、 少しずつ実装するとよいでしょう。

SimpleCalculator.java のコードを示します。

```
package work;

import java.awt.BorderLayout;
import java.awt.Container;//  入れ物のクラス

import javax.swing.*;

public class SimpleCalculator extends JFrame {
	Container container = getContentPane();
	SimpleCalcPanel panel = new NomalCalcPanel();

	public SimpleCalculator() {
		this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
		this.setTitle("SimpleCalculator");

		this.container.add( panel, BorderLayout.CENTER);

		//this.setResizable(false);//【1】ユーザがこのフレームのサイズを変更できなく設定します。

		this.setBounds(0, 0, 250, 250);//フレームサイズ指定
		this.setVisible(true);

		this.setAlwaysOnTop(true);//ほかのすべてのウィンドウの手前に表示されるように、最前面ウィンドウへ設定します
	}

	public static void main(String[] args) {
		JFrame f = new SimpleCalculator();
	}
}
```

上記で注目すべきは、SimpleCalcPanel の panel フィールドで NomalCalcPanel のオブジェクトを管理している点です。
これにより将来、演算処理が NomalCalcPanel と異なる操作を行なう（例えば 優先順位考慮など）クラスに変更する場合、SimpleCalcPanel の サブクラスであれば、簡単にここだけ変更すれば可能となりす。

以下に、SimpleCalcPanel.java のコードを示します。

```
package work;

import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.*;

public class SimpleCalcPanel extends JPanel implements ActionListener{
	JButton []btnNumb = new JButton[11];//  数字用
	JButton []btnOp = new JButton[6]; //オペレーション用ボタン
	JLabel lbl1 = new JLabel("0");
	JTextField txt1 = new JTextField("");

	public SimpleCalcPanel(){
		for(int i = 0; i  < btnNumb.length-1; i++){
			this.btnNumb[i] = new JButton("" + i);//数字表示ボタン生成
			this.add(this.btnNumb[i]);
		}
		this.btnNumb[10] = new JButton(".");
		this.add(this.btnNumb[10]);

		btnOp[0] = new JButton("+/-");
		btnOp[1] = new JButton("/");
		btnOp[2] = new JButton("*");
		btnOp[3] = new JButton("-");
		btnOp[4] = new JButton("+");
		btnOp[5] = new JButton("=");

		for(int i = 0; i  < btnOp.length; i++){
			this.add(this.btnOp[i]);//オペレーションボタン追加
		}
		this.add(this.lbl1);
		this.lbl1.setBackground(new Color(255, 255, 100));//ラベルの背景色設定
		this.lbl1.setOpaque(true);//ラベルを「不透明な」設定にします。
		this.lbl1.setHorizontalAlignment(SwingConstants.RIGHT);//右よせで文字列を表示
		this.lbl1.setFont(new Font(null, Font.BOLD, 24));

```

```
                this.add(this.txt1);
                this.txt1.setHorizontalAlignment(SwingConstants.RIGHT);//右よせで文字列を表示
                this.txt1.setFont(new Font(null, Font.BOLD, 24));
        }

        public void actionPerformed(ActionEvent e){
                //キー入力用ボタン処理記述予定
        }
}
```

初期の確認用コードとしては、このように、部品を配置しているだけでよいでしょう。
SimpleCalculator のサイズを固定にする場合は、この にチェックを付けてください。これで【1】のコメントのコードが有効になります。

以下に、この時の NomalCalcPanel.java コードを示します。

```
package work;

import java.awt.event.*;
import javax.swing.*;

public class NomalCalcPanel extends SimpleCalcPanel // implements ActionListener
{
//        public void actionPerformed(ActionEvent e){
//演算用用ボタン処理記述予定
//        }
}
```

SimpleCalcPanel の ActionListener 実装は、後で行う予定なので、始めはコメントにしてください。

◆Swing を使ったアプリケーションとは
https://www.javadrive.jp/tutorial/ini/index1.html

＊レイアウトマネージャーによるコンポーネントの配置
ボタンやラベルなどのコンポーネントをフレームなどの追加していく場合、
Swing ではレイアウトマネージャーと呼ばれるものを使います。
細かいサイズや位置を指定するのではなく、複数用意されたレイアウトマネージャーの中から希望する画面構成にあったものを選び、
実際の配置はレイアウトマネージャーに任せます。

例えばフレームの下部にボタンを横一列に並べたい場合には
FlowLayout と呼ばれるレイアウトマネージャーが適しています。

FlowLayout はコンポーネントを追加した順に右へ右へと追加していくレイアウトマネージャーです。
またこのレイアウトマネージャーは追加されるコンポーネントのサイズを自動調整しません。

```
import javax.swing.*;
import java.awt.*;

class JSample1_4{
   public static void main(String args[]){
      JFrame frame = new JFrame("MyTitle");
      frame.setBounds(100, 100, 600, 400);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      JPanel p = new JPanel();
      JButton btn1 = new JButton("Save");
      JButton btn2 = new JButton("Cancel");
      JButton btn3 = new JButton("Help");

      p.add(btn1);
      p.add(btn2);
      p.add(btn3);

      frame.getContentPane().add(p, BorderLayout.SOUTH);
      frame.setVisible(true);
   }
}
```

実行してみると 3 つのボタンが横に並んで表示されています。