

デジタル表現論 2016

第9回 Swingコンポーネントを使う

担当: 劉 雪峰

復習と今回の内容

□ 前回の内容

- Javaで色を設定する
- 数学のグラフを描く

□ 今回の内容

- ラベル・テキストフィールドを利用する.
- ボタンを利用する.

レポート課題

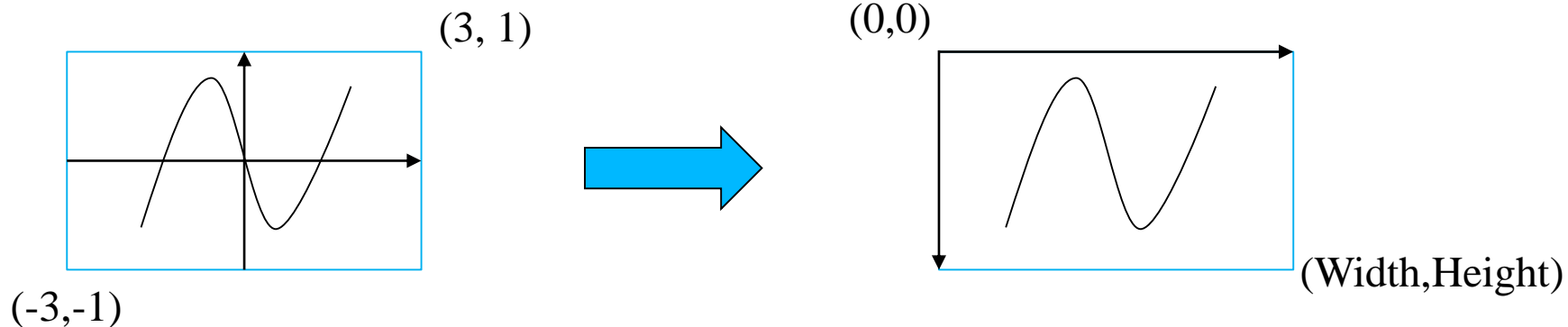
座標変換

□ グラフの範囲

$$[-3,3] \times [-1,1]$$

□ コンピュータの範囲 (高さ: Height、幅: Width)

$$[0, \text{Width}] \times [0, \text{Height}]$$

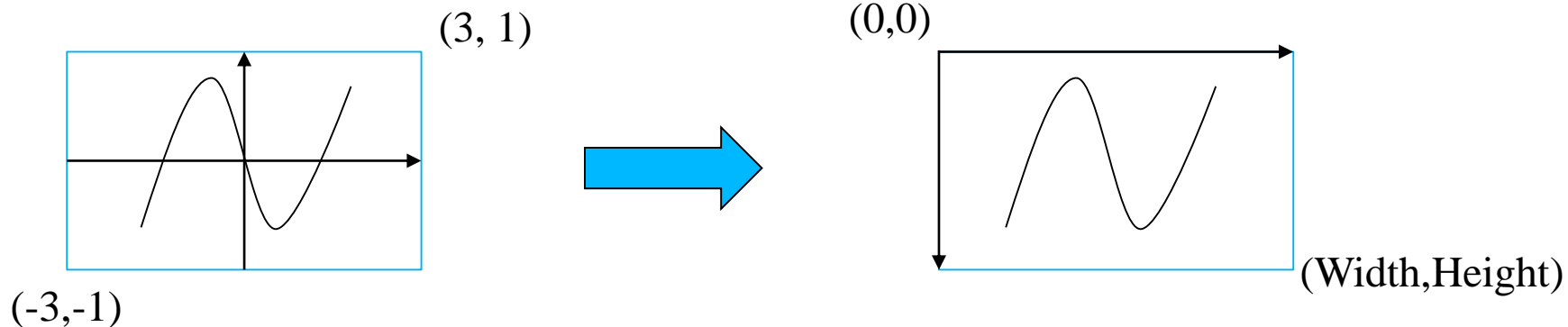


より一般的な座標変換

座標変換 $(x,y) \Rightarrow (px,py)$

- グラフの範囲: $[xmin,xmax] \times [ymin,ymax]$
- コンピュータの範囲 $[0,Width] \times [0,Height]$

$$px = (x-xmin)/(xmax-xmin)*Width$$
$$py = Height - (y-ymin)/(ymax-ymin)*Height;$$



Javaコードの作成

- 新たなメソッドを定義する。(HPの解答例を参考しなさい。)

```
//x座標の変換
```

```
public double x_change(double x, double width, double x_min, double x_max){  
    double new_x;  
    new_x = (x - x_min)/(x_max - x_min) * width;  
    return new_x;  
}
```

```
//y座標の変換
```

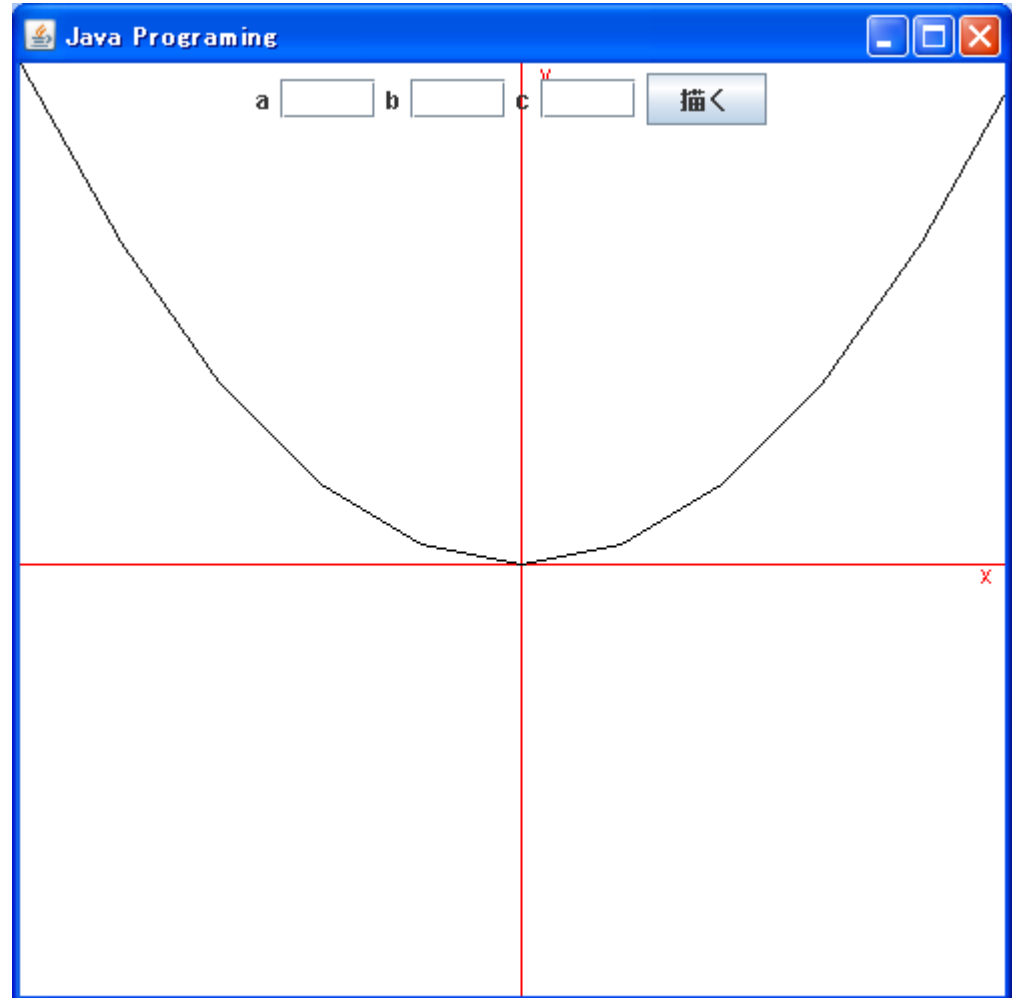
```
public double y_change(double y, double height, double y_min, double y_max){  
    double new_y;  
    new_y = (y - y_min)/(y_max - y_min) * height;  
    new_y = height - new_y;  
    return new_y;  
}
```

本日の目標

- 本日は $y=ax^2+bx+c$ のグラフを書かせます.
 - ただし, 係数 a, b, c は画面上から入力させて実行させるようにします.
- (本日の雛形Hina9.javaとその説明)

イメージ

例えばこの
画面を作り
実行させます。



Swing・AWTについて

- GUI (Graphical User Interface)を作成するためのJavaのライブラリを用いる.
- **AWT (Abstract Windowing Tools):**
GUIアプリケーションを作成するためのクラスライブラリ
- **Swing:**
Awtを継承した軽量コンポーネント

演習

- HPからHina9_0.javaダウンロードして実行してみてください。
- 画面に書かれたボタンなどを確認してください。

JTextField

- 画面上に文字を入力させるスペースが必要です(今回は3つ).
- それにはJTextFieldクラスを利用します.

□ 参考

<http://docs.oracle.com/javase/jp/6/api/javax/swing/JTextField.html>

JTextField

□ 宣言：`public class MyJPanel`の後に

`JTextField textField1, textField2, textField3;`

□ 設定（インスタンスの生成）：

`public MyJPanel(){...}` の中に以下のコードを入れる。

`textField1 = new JTextField("1.0",4);`

`textField2 = new JTextField("0.0",4);`

`textField3 = new JTextField("0.0",4);`

Label

- 画面に3つの入力欄が現れましたが、それぞれの役割の説明が欲しいところです。
- そこで、ラベルというものを使いましょう。
- 宣言 : `public class MyJPanel`の後に
`JLabel label1, label2, label3;`

Label

ラベルを画面に貼るには JLabel クラスを使用します。宣言と配置は以下の通りにします。

```
label1 = new JLabel("a");
```

```
label2 = new JLabel("b");
```

```
label3 = new JLabel("c");
```

```
add(label1); add(label2); add(label3);
```

TextFieldとLabelの配置

以下の命令により, 画面にラベルと入力欄を配置することができます。

- `add(label1);` //画面にラベルが登場します.
- `add(textField1);` //画面に入力欄が登場します.

LabelとTextFieldの配置

実際のコードでは、以下のように

```
add(label1);
```

```
add(textField1);
```

```
add(label2);
```

```
add(textField2);
```

```
add(label3);
```

```
add(textField3);
```

と順番を気にしておきましょう。

レイアウトについて

- 例えば下記のように2行にわたって配置したいときは

a:

b:

c:

Java GridLayoutやBoxLayoutなどをキーワードに調べてみてください。

(Layoutの指定法はもっとたくさんあります。)

演習2

- ここまでの説明に従って、画面にlabel, textfiledを追加してみてください。

ボタンを使用する

- 数値を入力した後に、グラフを描かせるボタンを作成しましょう.
- 使用するクラスはJButtonクラスを用います.
- 参考:

<http://docs.oracle.com/javase/jp/6/api/javaw/swing/JButton.html>

ボタンを使用する

□ 宣言:

```
JButton beginButton;
```

□ 設定(インスタンス):

```
beginButton = new JButton("描く");
```

□ 画面に加える:

```
add(beginButton);
```

ActionListenerについて

- ActionListenerに登録されているコンポーネントの動作を確認したら、ActionPerformedメソッドを呼び出す機能を持つ。
- `public implements ActionListener`
このクラスで ActionListenerを使用することを明記する。
- `beginButton.addActionListener(this);`
このボタンをActionListenerに登録する。

イベント処理

```
public void actionPerformed(ActionEvent e) {  
    //この中にイベント処理の内容を書く  
}
```

- このメソッドはボタンが押されたら自動的に呼びだされる(正確にはactionListenerに登録されているものが実行されたとき)。

イベント処理

```
public void actionPerformed(ActionEvent e) {  
    if(e.getSource()==beginButton)  
    {  
        a=Double.parseDouble(textField1.getText());  
        b=Double.parseDouble(textField2.getText());  
        c=Double.parseDouble(textField3.getText());  
        repaint();  
    }  
}
```

- (注意) double a,b,c;をMyJPanelの次に書いておく.

演習3

- ここまでの説明に従って、画面にbuttonを追加してみてください。

本題

□ $y=ax^2+bx+c$ のグラフを書かせましょう.

雛形の

```
x1=x;
```

```
y1=x1*x1;
```

```
x2=(x+incx);
```

```
y2=x2*x2;
```

の部分をa, b, cを用いて書き直しましょう.

a,b,cの宣言はmyJPanelの前に行う.

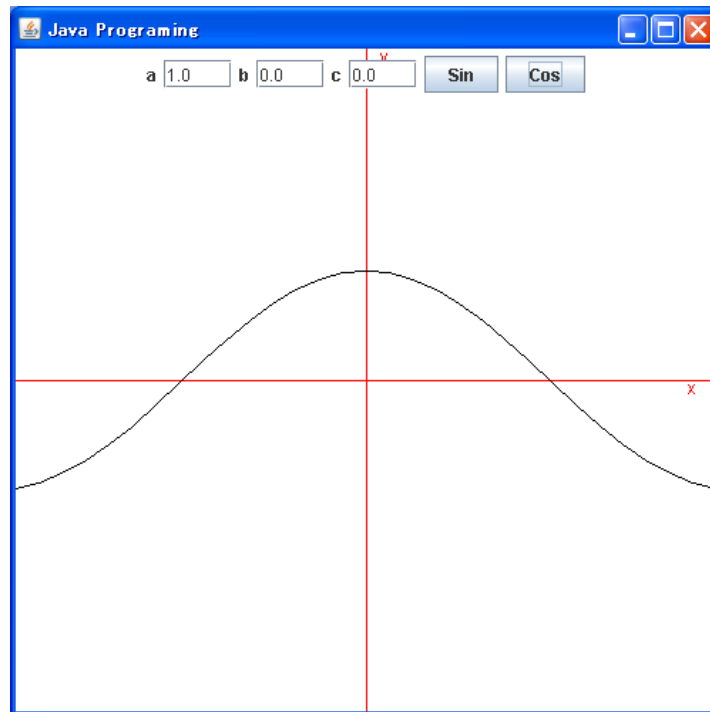
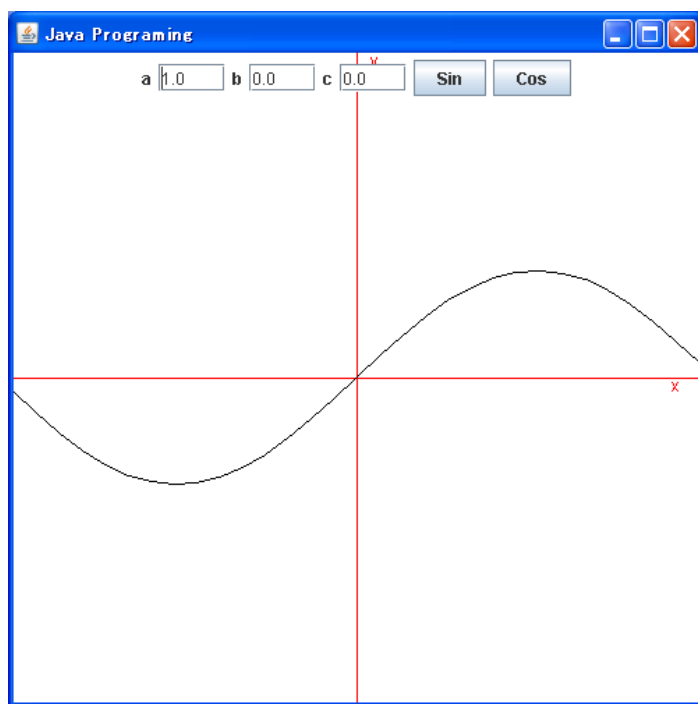
(雛ファイル Hina9.javaを利用して作業すること)

本日の自習課題1

- $y=a \sin(bx+c)$ の a と b と c の3つの数値を入力してもらい、このグラフを表示しなさい.
- ただし、 x の変域は $-3 \leq x \leq 3$ とします.
(グラフを描かせる部分も変更が必要です)
(変域もJTextFieldで入力させて自動で描かせるよりも良い！)

本日の自習課題2

- ボタンを2つ用意し, 1つのボタンではsin関数を, もう1つのボタンではcos関数を描かせるように使いわけなさい.



本日の自習課題3(オプション)

- 自習課題2であげたボタンによりSin関数かCos関数かを描かせるプログラムをJRadioButtonを用いて作成しなさい.
- (つまりJRadioButtonについて自ら調べてみてくださいということです.)