

# Java プログラミング入門

— グラフィックの描画：ラベル・テキスト —

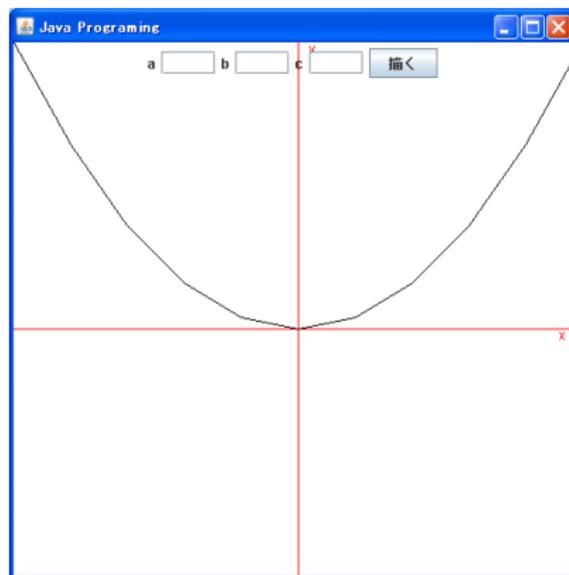
早稲田大学

# 本日の目標

## 以下の GUI を完成させる

関数  $y = ax^2 + bx + c$  のグラフを描く．ただし，係数  $a, b, c$  は画面上から入力して実行する．

イメージ：



# Swing・AWT について

## GUI を作成

- GUI (Graphical User Interface) を作成するため、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。AWT を拡張した GUI ツールキットである

# JTextField

## JTextField クラス

- 画面上に文字を入力させるスペースを作成する  
(今回は3つ)
- JTextField クラスを利用する

参考 :

<https://docs.oracle.com/javase/7/docs/api/javax/swing/JTextField.html>

## JTextField クラス

- 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

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

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

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

## textField (おまじない)

```
add(textField1);
```

```
add(textField2);
```

```
add(textField3);
```

を入力。これにより、画面に3つの入力欄が配置される。

# Label

## Label

- JTextField を利用して画面に 3 つの入力欄が現れた
- 次にそれぞれの役割を説明するためにラベルを使用する

- 宣言 :

```
public class MyJPanel の後に
```

```
    JLabel label1, label2, label3;
```

- ラベルを画面に貼るには JLabel クラスを使用する .
- 宣言 :

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

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

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

## Label

- 以下の命令により，画面にラベルを配置

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

- **注意**：

```
add(label1);  
add(textField1);  
add(label2);  
add(textField2);  
add(label3);  
add(textField3);
```

の順番に注意！

# ボタンを使用する

## ボタン

- 数値を入力した後に、グラフを書かせるボタンを作成する
- 使用するクラスは JButton クラス
- JButton クラスを調べよう

参考：<https://docs.oracle.com/javase/jp/6/api/javax/swing/JButton.html>

# ボタンを使用する

## ボタンの仕様

- 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

- 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がある
- `public **implements ActionListener`  
クラスで ActionListener を使用することを明記
- `beginButton.addActionListener(this);`  
このボタンを ActionListener に登録

# イベント処理

## 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

## 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 .

# 例題

## 以下のグラフをかけ

$y = ax^2 + bx + c$  のグラフをかけ．ただし，係数  $a, b, c$  は画面上から入力させて実行させる．

- 雛形の

```
x1=x;
```

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

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

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

の部分をも  $a, b, c$  を用いて書き直す．

- $a, b, c$  の宣言は `myJPanel` の前に行く．