



SIMPLEPLOTTERのヒント

オブジェクト指向プログラミング特論

只木進一:総合情報基盤センター

SIMPLE PLOTTER 作成のヒント

- データ読み込み
- 作図パラメタ
- プロットの実際
- 描画範囲



データ読み込みのヒント

- 入力した文字列をスペースで区切る
 - `String[] String.split(String reg)`の利用
 - `reg`:正規表現
- 数値が読み出せない場合への対策
 - `NumberFormatException`をcatchする



```
String line;
String nl = System.getProperty("line.separator");//改行コード

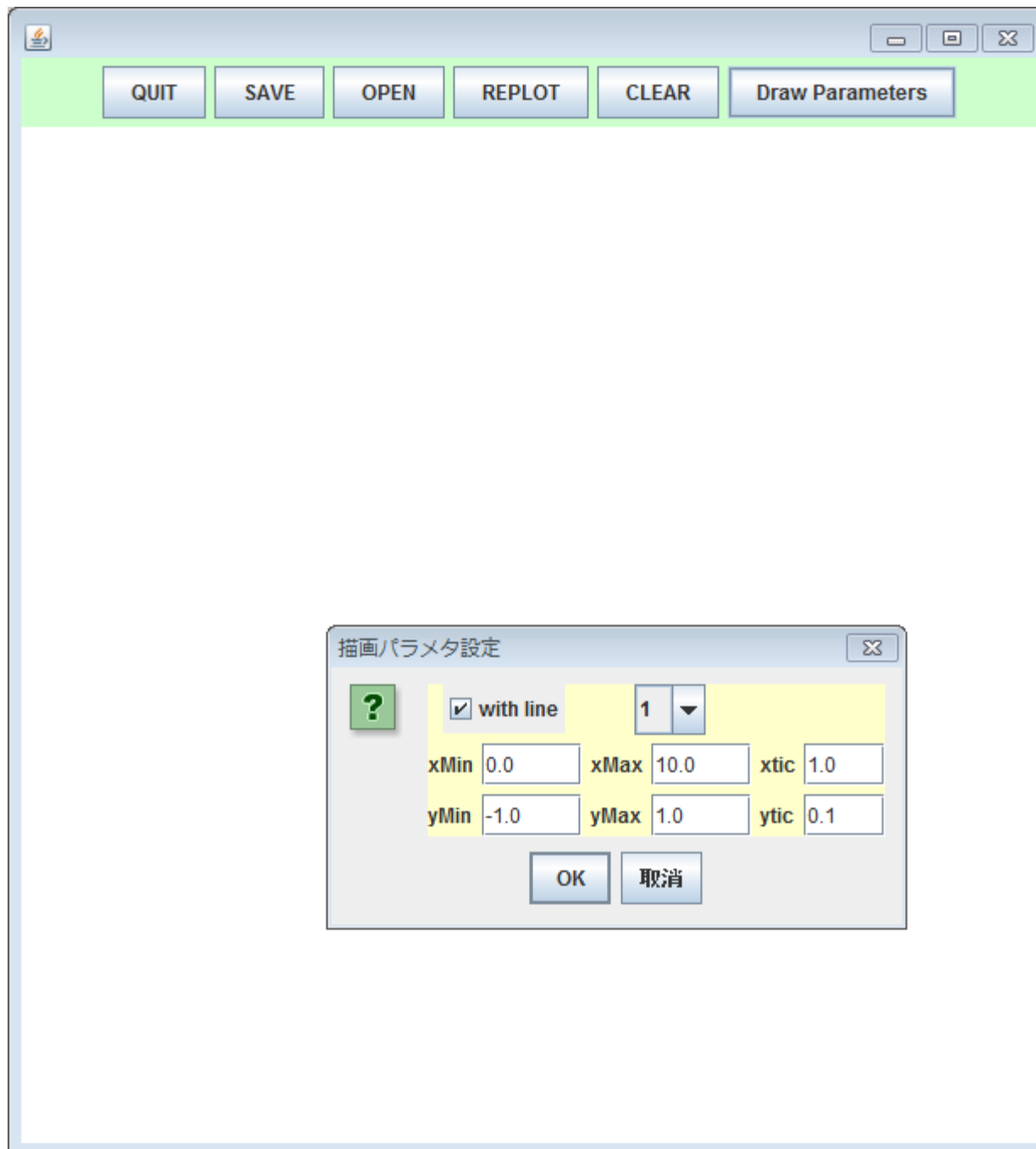
while ((line = in.readLine()) != null) {
    String s[] = line.split("¥¥s+");
    double x = 0, y = 0;
    try {
        x = Double.valueOf(s[0]);
        y = Double.valueOf(s[1]);
    } catch (NumberFormatException numberFormatException) {
    }
    //読み込んだデータを使う
}
```



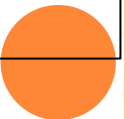
作図パラメタの設定

- 作図パラメタをひとまとめにしてクラス設計
 - X軸Y軸の、最小、最大
 - 線を引くか否か
 - 線の太さ・色
- パラメタを設定するフォームをJPanelで作る
- JOptionPaneとして表示





```
private void setParameterActionPerformed(java.awt.event.ActionEvent evt) {  
    drawParameterPanel.showParameter();  
    int answer = JOptionPane.showOptionDialog(  
        new JFrame(), drawParameterPanel,  
        "描画パラメタ設定", JOptionPane.OK_CANCEL_OPTION,  
        JOptionPane.QUESTION_MESSAGE, null, null, null);  
  
    if (answer == JOptionPane.OK_OPTION) {  
        drawParameterPanel.updateParameter();  
        drawPanel.reDrawData();  
        drawPanel.repaint();  
    }  
}
```



プロットの実際

- 座標系への注意
 - Y軸が下向き
- 縮尺の設定
 - BufferedImageのサイズと作図範囲
 - BufferedImage.getWidth(), BufferedImage.getHeight()
- 原点の移動
 - Graphics2D.translate(double x,double y)
 - BufferedImageのサイズ、マージン



- 折れ線を描く
 - Path2D.Double の利用
 - Path2D.Double.moveTo(x,y)
 - Path2D.Double.lineTo(x,y)

```
Path2D.Double path = new Path2D.Double();
path.moveTo(factor.x * data.get(0).x, -factor.y * data.get(0).y);
for (int i = 1; i < data.size(); i++) {
    path.lineTo(factor.x * data.get(i).x, -factor.y * data.get(i).y);
}
g.setColor(getForeground());
g.draw(path);
```



描画範囲の設定

- データの最大値と最小値の絶対値の大きいほうに注目
- 絶対値の最上位桁
 - `int k = (int)Math.log10(x)`

