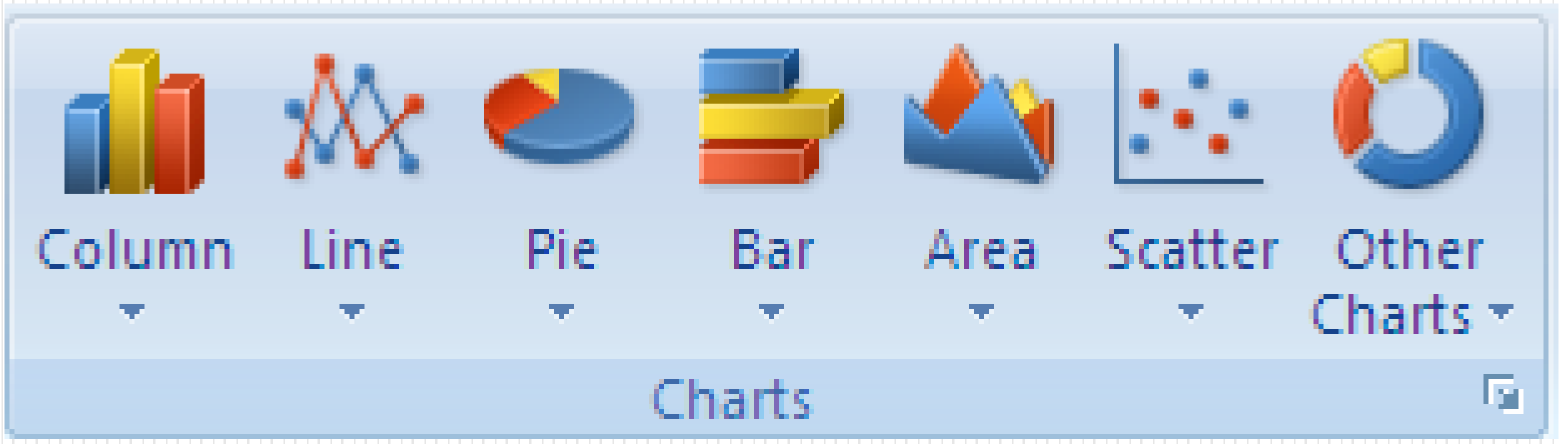
Lecture 05
Two Dimensional Graphics
Part 4

Visualizing Real Data



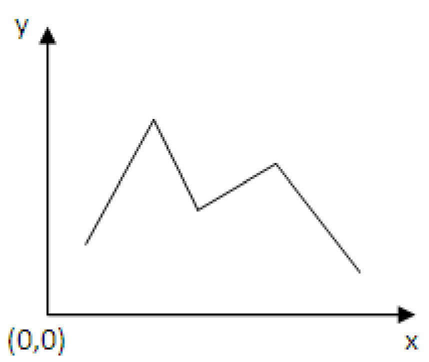
Introduction

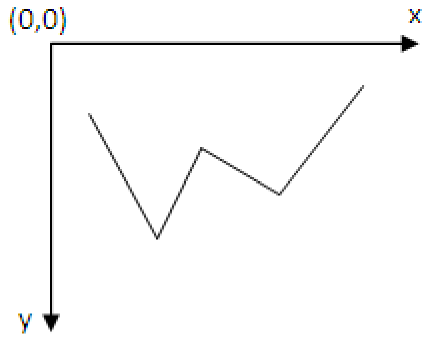
- Visualizing data is a common request in some applications.
- Graphical data is easier to stand than text
- Programmers should have the skill at data visualization
- Demo: Creating Line Chart for a time series

Contents

- Normal Coordinates versus Device Coordinates
- Expected Shape
- Mapping a Real Point to Device Point
- Drawing Area in a Component

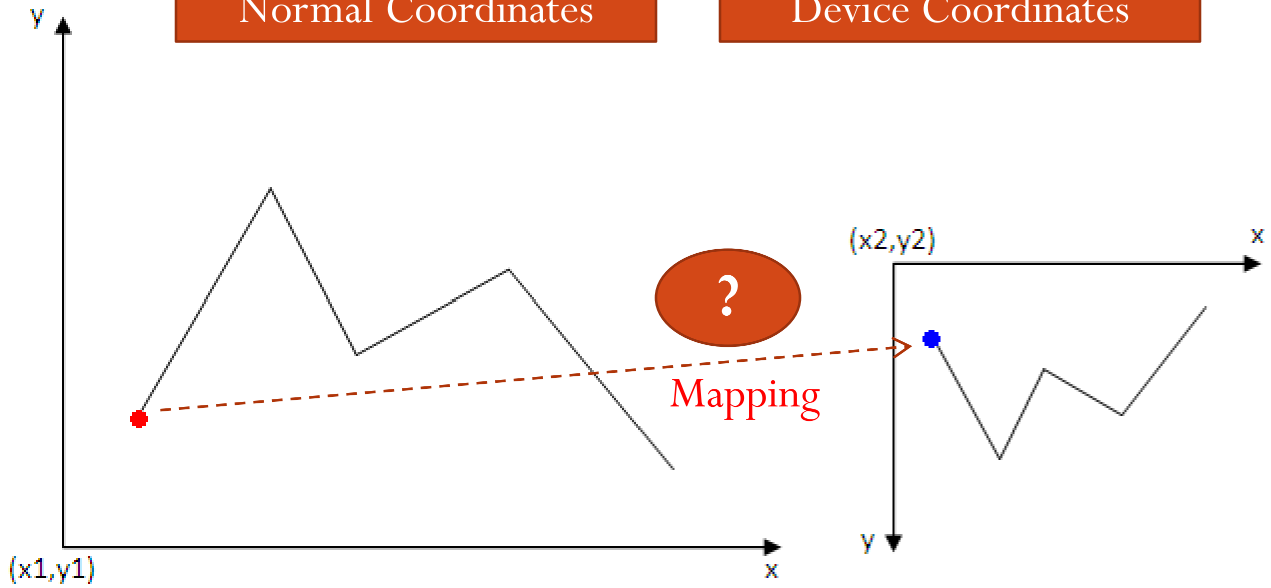
Normal Coordinates versus Device Coordinates





Normal Coordinates

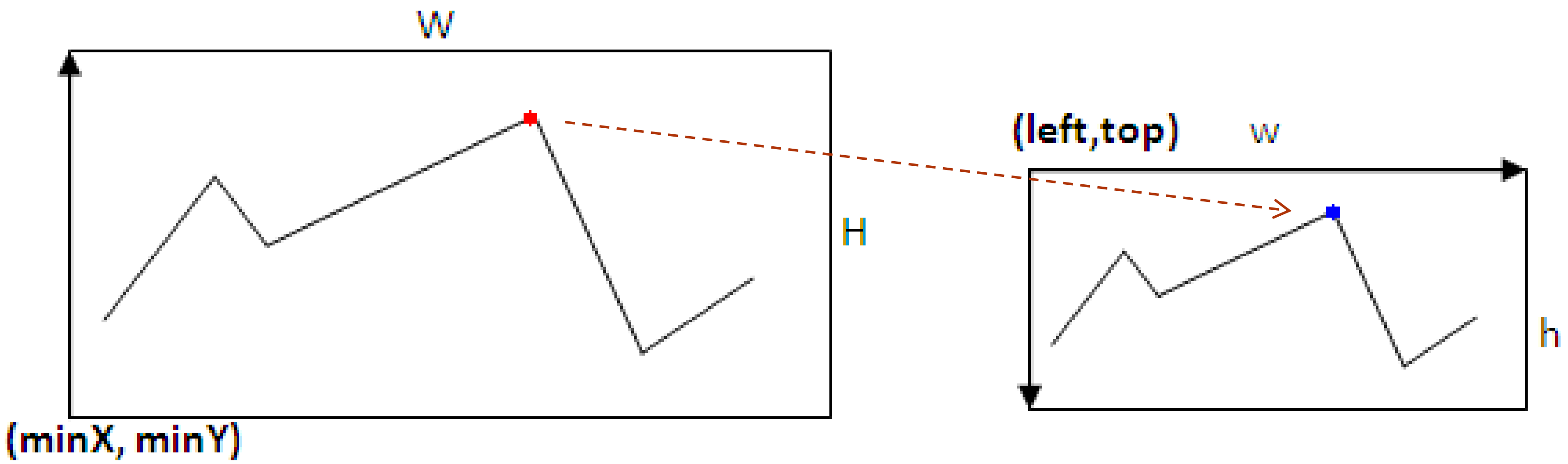
Device Coordinates



Normal Coordinates

Device Coordinates

Expected Shape

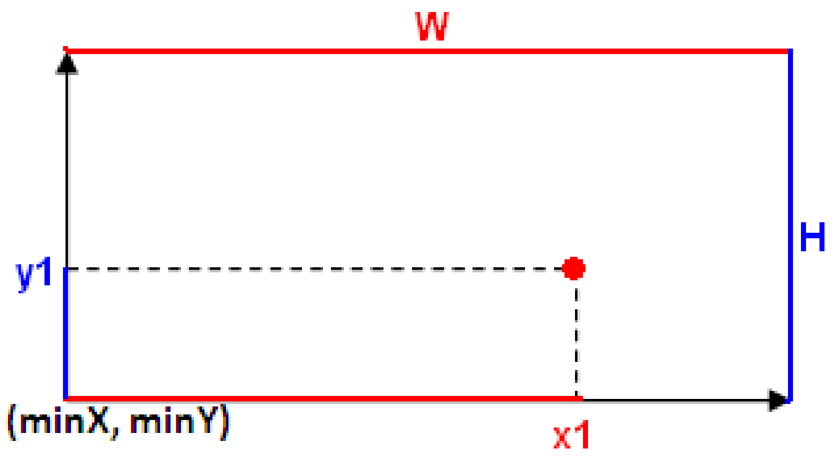


Common used shape

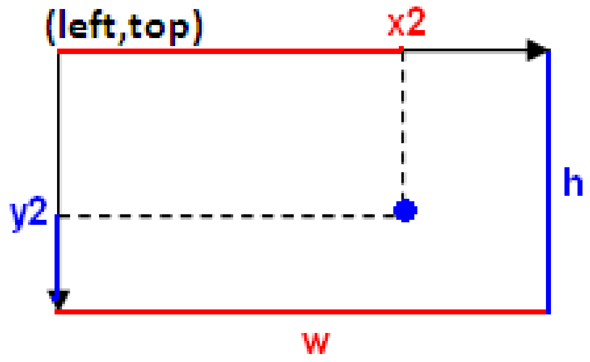
The shape will be on device

A mapping is needed to exchange a real point to an on-device point

Mapping a Real Point to Device Point



Normal Coordinates



Device Coordinates

Properties of the point must be preserved

Computing x2

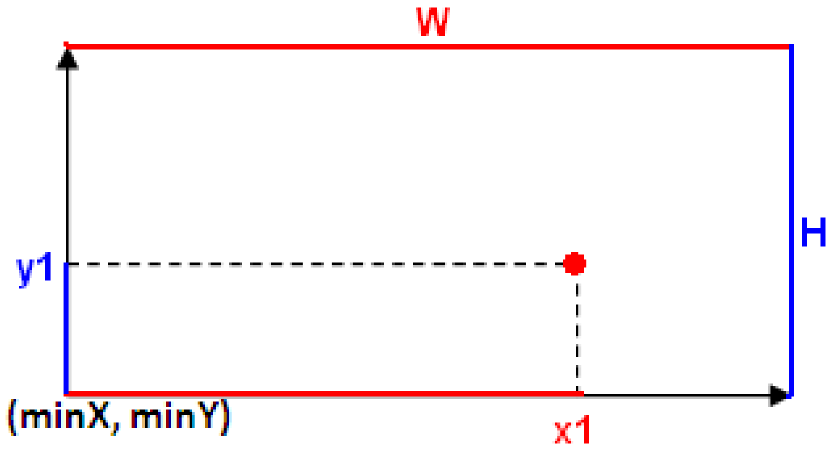
(x1-minX)/W = (x2-left)/w → x2 = left + (x1-minX)*w/W

Let $C_W = w/W$

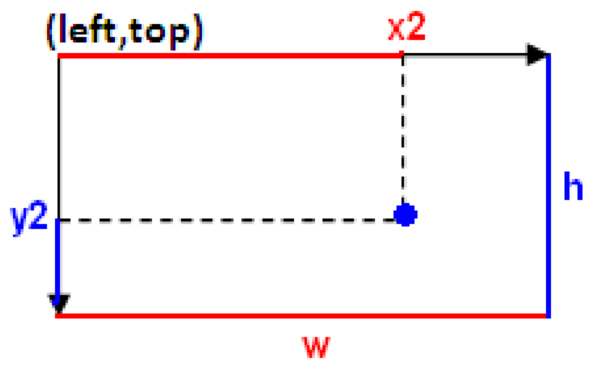
→x2 = left + C_W(x1-minX)

→x2 = C_wx1 + left – $C_w$minX

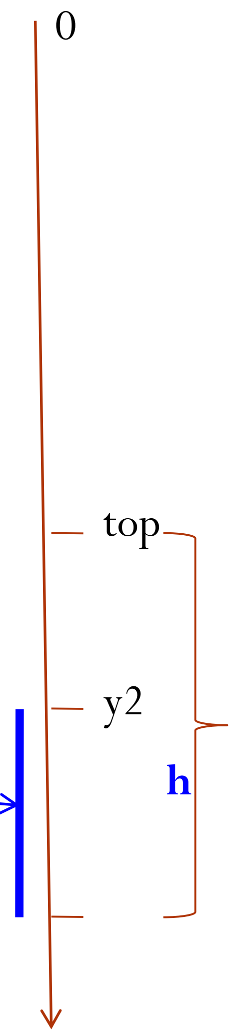
Mapping a Point



Normal Coordinates



Device Coordinates



Properties of the point must be preserved

Computing y2

(y1-minY)/H = (top+h-y2)/h

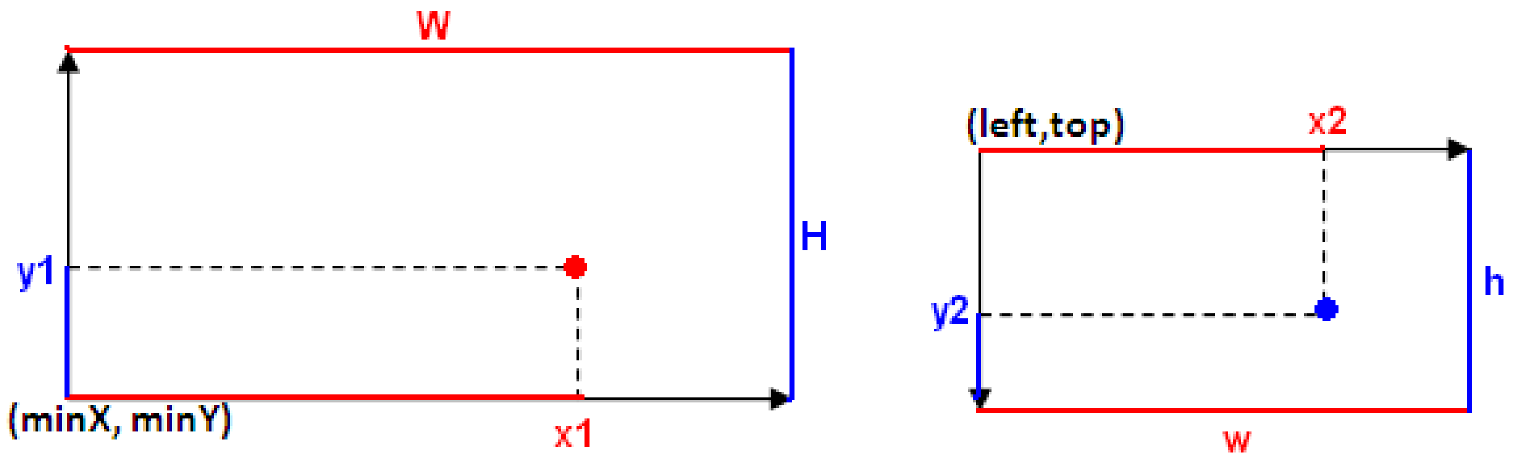
(y1-minY)*h/H = top +h –y2

Let $C_h = h/H$

➔y2 = top + h – C_h(y1-minY)

➔y2 = -C_hy1 + $C_h$minY + top + h

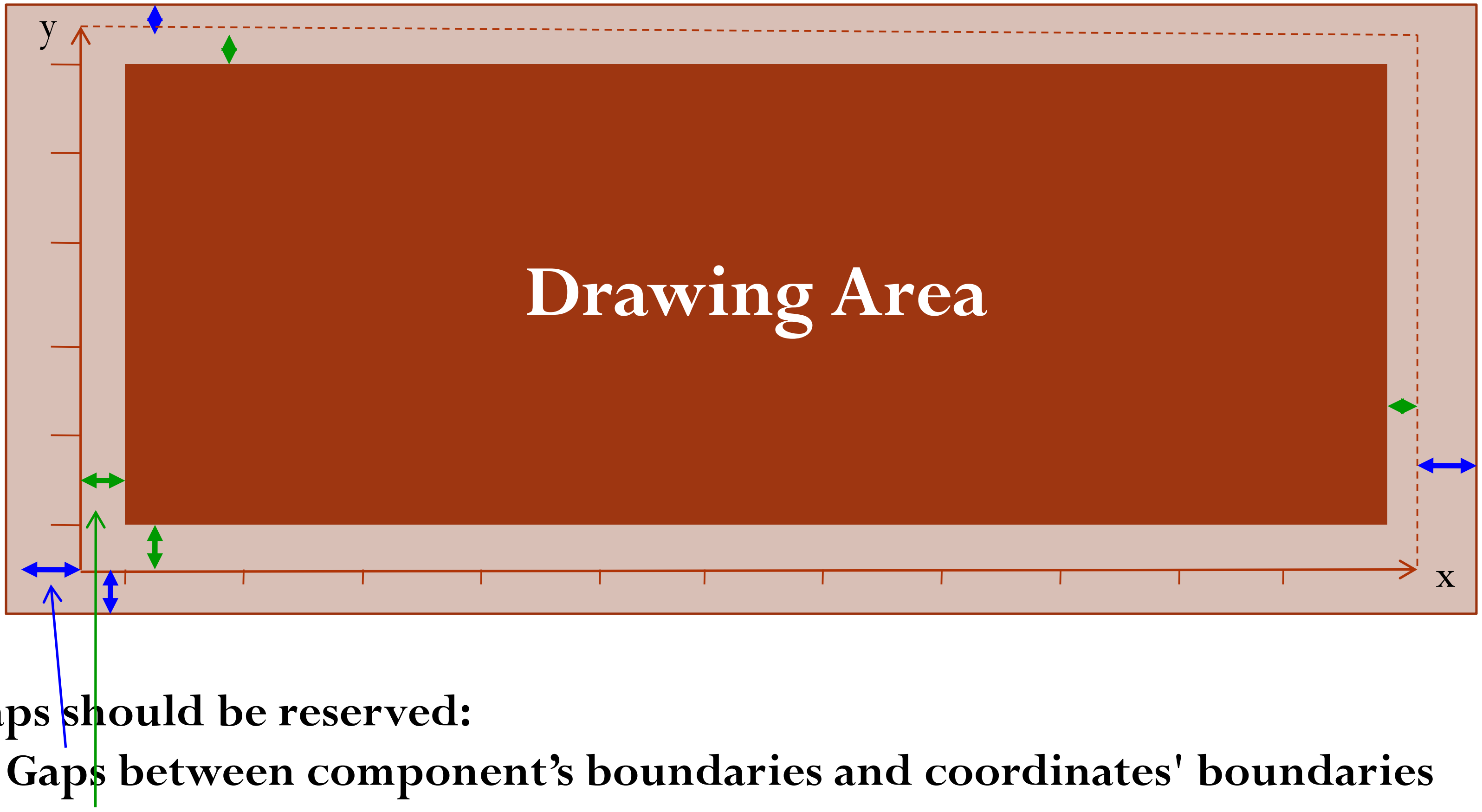
Mapping a Point



- $x2 = C_w x1 + \underline{left - C_w minX}$
- $x2 = C_w x1 + C1$
- $y2 = - C_h y1 + \underline{C_h minY + top + h}$
- $y2 = - C_h y1 + C_2$

Classes can be use in Java API:
-java.awt.**Dimension**: An area on component
-java.awt.geom.**AffineTransform for mapping a point in one window to another**

Drawing Area in a Component



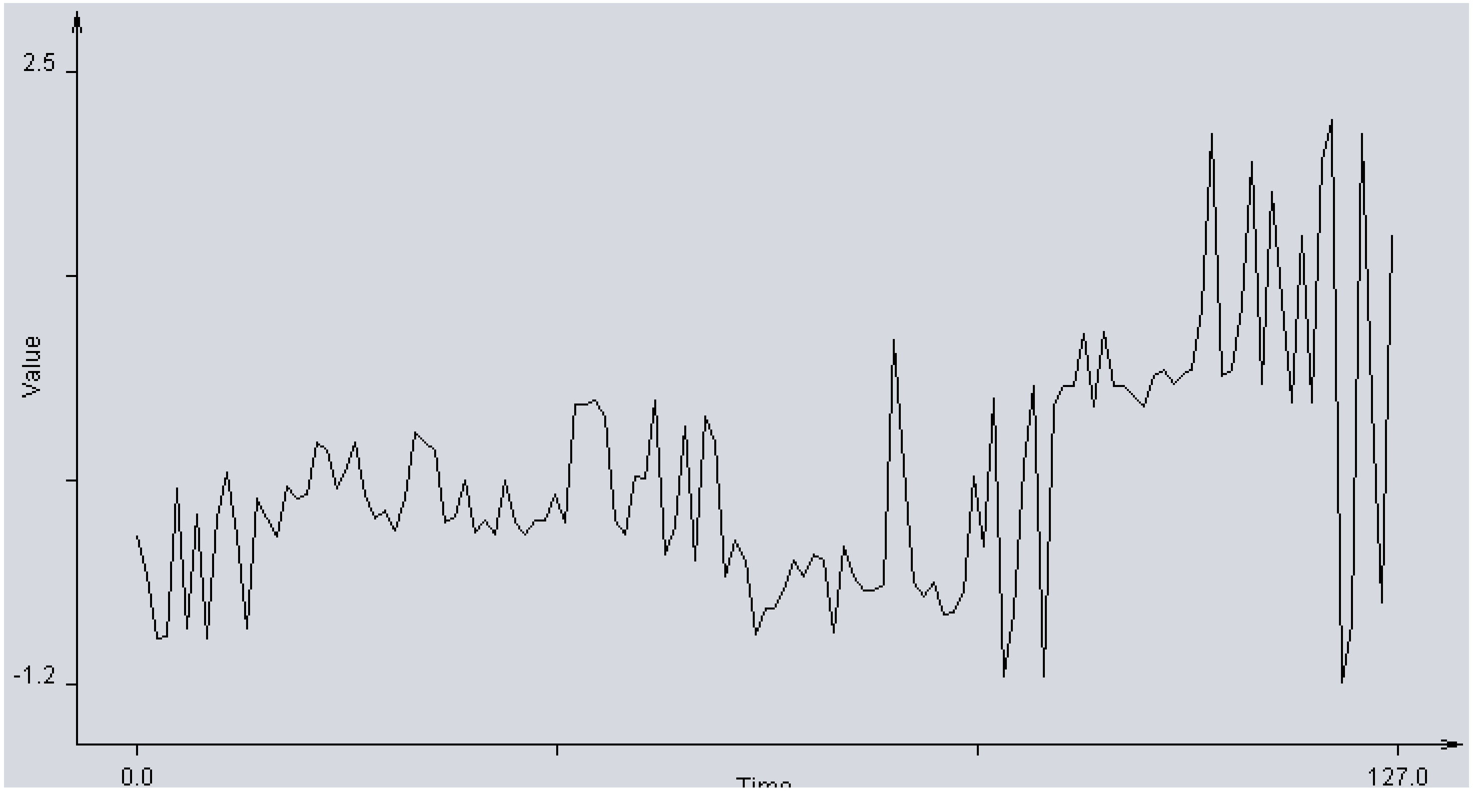
Gaps should be reserved:

Gaps between component's boundaries and coordinates' boundaries

Gaps between coordinates' boundaries and drawing boundaries

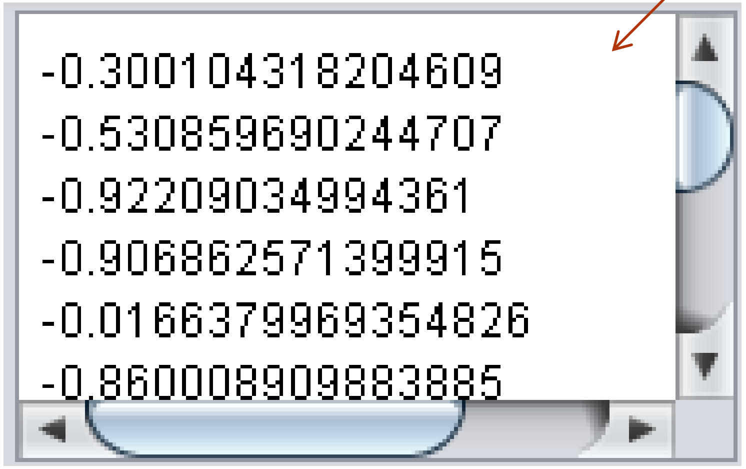
Demo: Creating Line Chart

- A **time series** is a sequence of data points, typically consisting of successive measurements made over a time interval. Examples of time series are electrocardiograms (ECG), ocean tides, counts of sunspots, and the daily closing value of the Dow Jones Industrial Average. Time series are very frequently plotted via line charts. (Wikipedia)



Demo…

- This demonstration depicts how to draw a line chart for a time series.
- Time series is usually stored as a list of values, time intervals are default. Example:

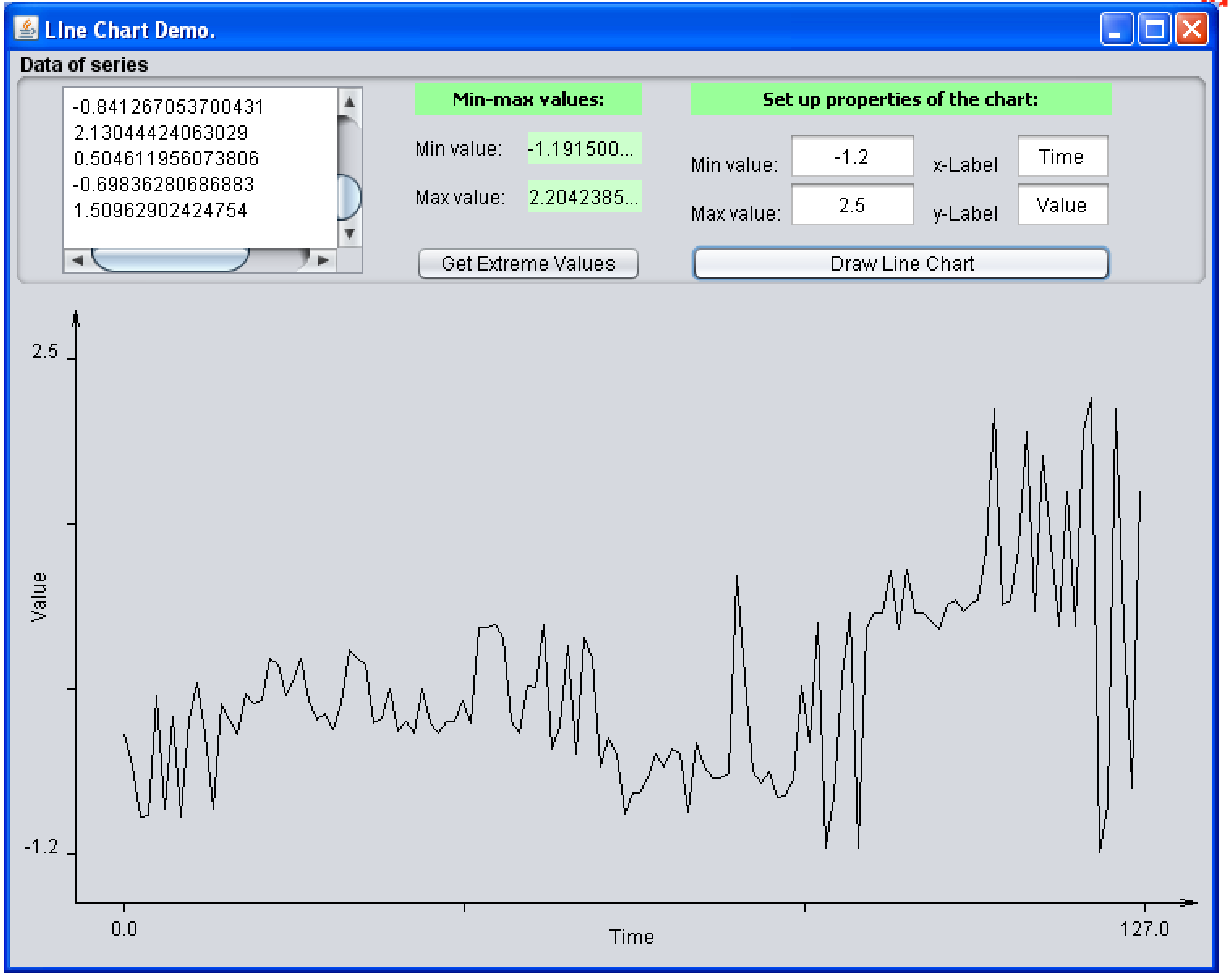


```
-0.300104318204609
-0.530859690244707
-0.92209034994361
-0.906862571399915
-0.0166379969354826
-0.860008909883885
```

They can be considered as points:
(0, -0.300104318204609)
(1, -0.530859690244707)
(2, -0.92209034994361)
(3, -0.906862571399915)
….

Demo …

GUI



```
-0.841267053700431
2.13044424063029
0.504611956073806
-0.69836280686883
1.50962902424754
```

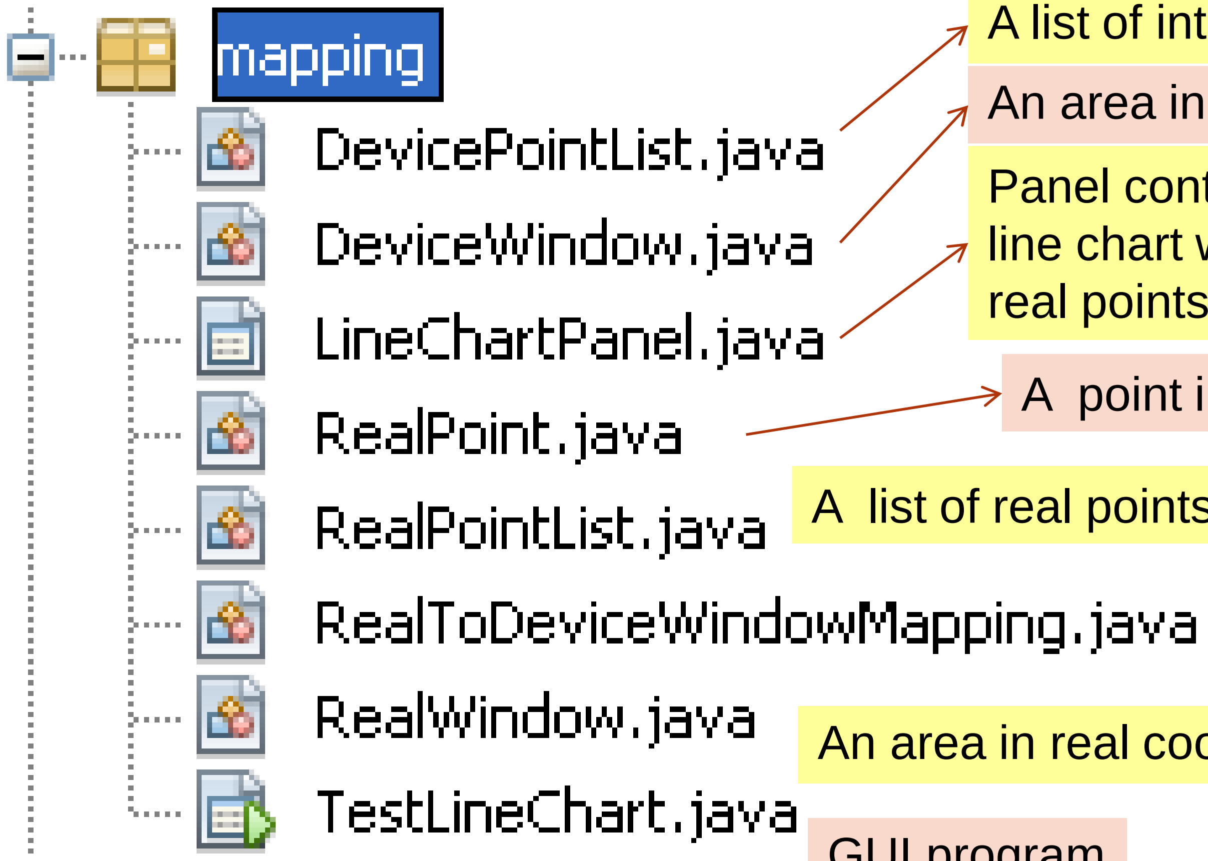
Min value: -1.191500...
Max value: 2.2042385...

Min value: -1.2
Max value: 2.5
x-Label: Time
y-Label: Value

TRƯỜNG ĐẠI HỌC FPT

Demo …

Class Design



mapping

DevicePointList.java

DeviceWindow.java

LineChartPanel.java

RealPoint.java

RealPointList.java

RealToDeviceWindowMapping.java

RealWindow.java

TestLineChart.java

A list of integral points on a device

An area in device coordinates

Panel contains ability for drawing a line chart which representing a list of real points

A point in real coordinates

A list of real points

Mapping a real window to a device window

An area in real coordinates

GUI program

TRƯỜNG ĐẠI HỌC FPT

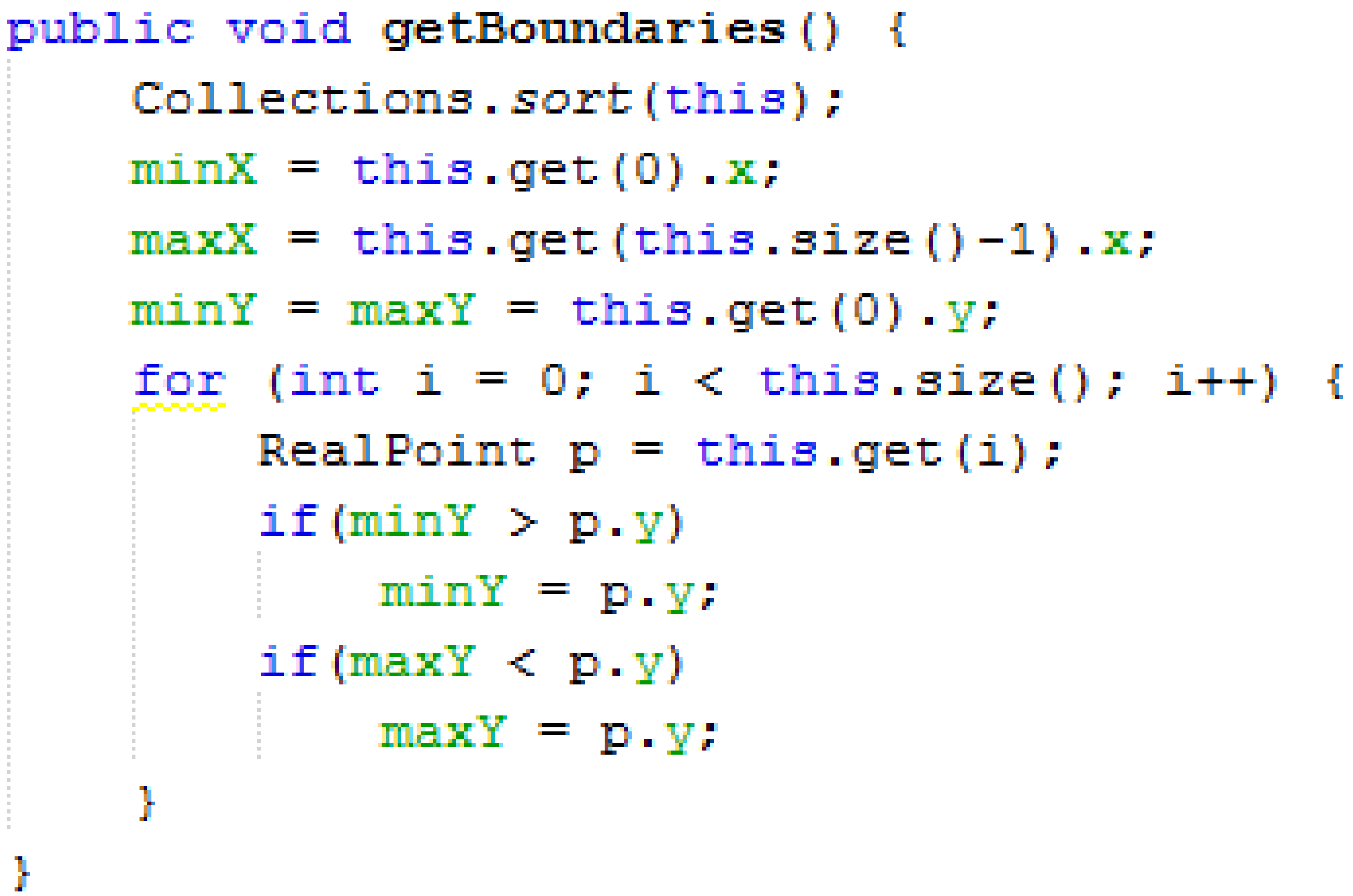
Demo.: Code

```
/* Class for real points */
package mapping;
public class RealPoint implements Comparable<RealPoint> {
    public double x, y;
    public RealPoint(double x, double y) {
        this.x = x;
        this.y = y;
    }
    @Override
    // Function for camparison based on x-coordinate
    public int compareTo(RealPoint p) {
      int result =0;
      if (this.x > p.x) result = 1;
      if (this.x < p.x) result = -1;
      return result;
    }
}
```

Demo.: Code

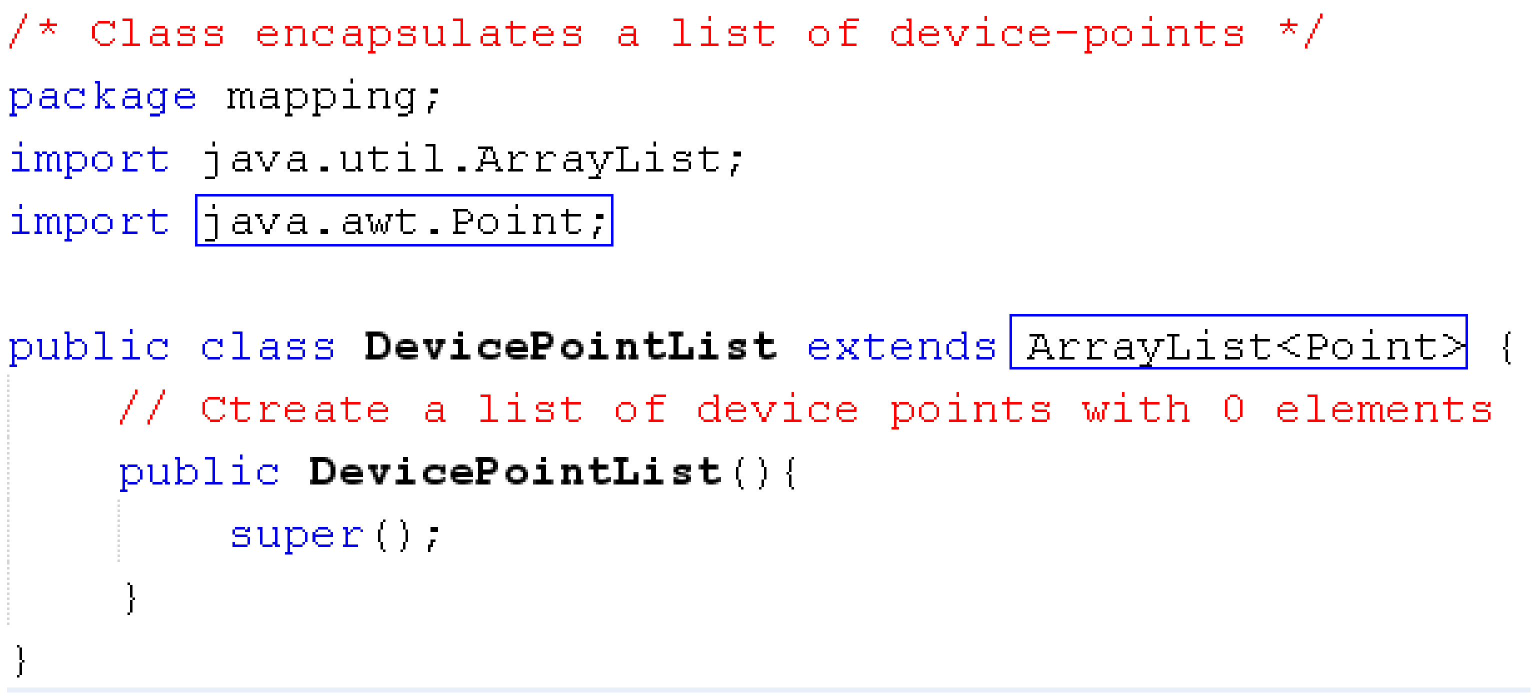
```
/* Class encapsulates a list of real Points */
package mapping;
import java.util.Collection;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Collections;
public class RealPointList extends ArrayList<RealPoint> {
    public double minX, minY;
    public double maxX, maxY;
    public RealPointList() {
        super();
    }
    // Create a list of real points from array of values (y-coordinates)
    // Values at the x-axis are default values of 0, 1, 2,...
    public RealPointList (double[] values ){
        super();
        for (int i=0; i<values.length; i++ )
            this.add(new RealPoint(i, values[i]));
        getBoundaries();
    }
```

Demo.: Code



```
public void getBoundaries() {
    Collections.sort(this);
    minX = this.get(0).x;
    maxX = this.get(this.size()-1).x;
    minY = maxY = this.get(0).y;
    for (int i = 0; i < this.size(); i++) {
        RealPoint p = this.get(i);
        if(minY > p.y)
            minY = p.y;
        if(maxY < p.y)
            maxY = p.y;
    }
}
```

Demo.: Code

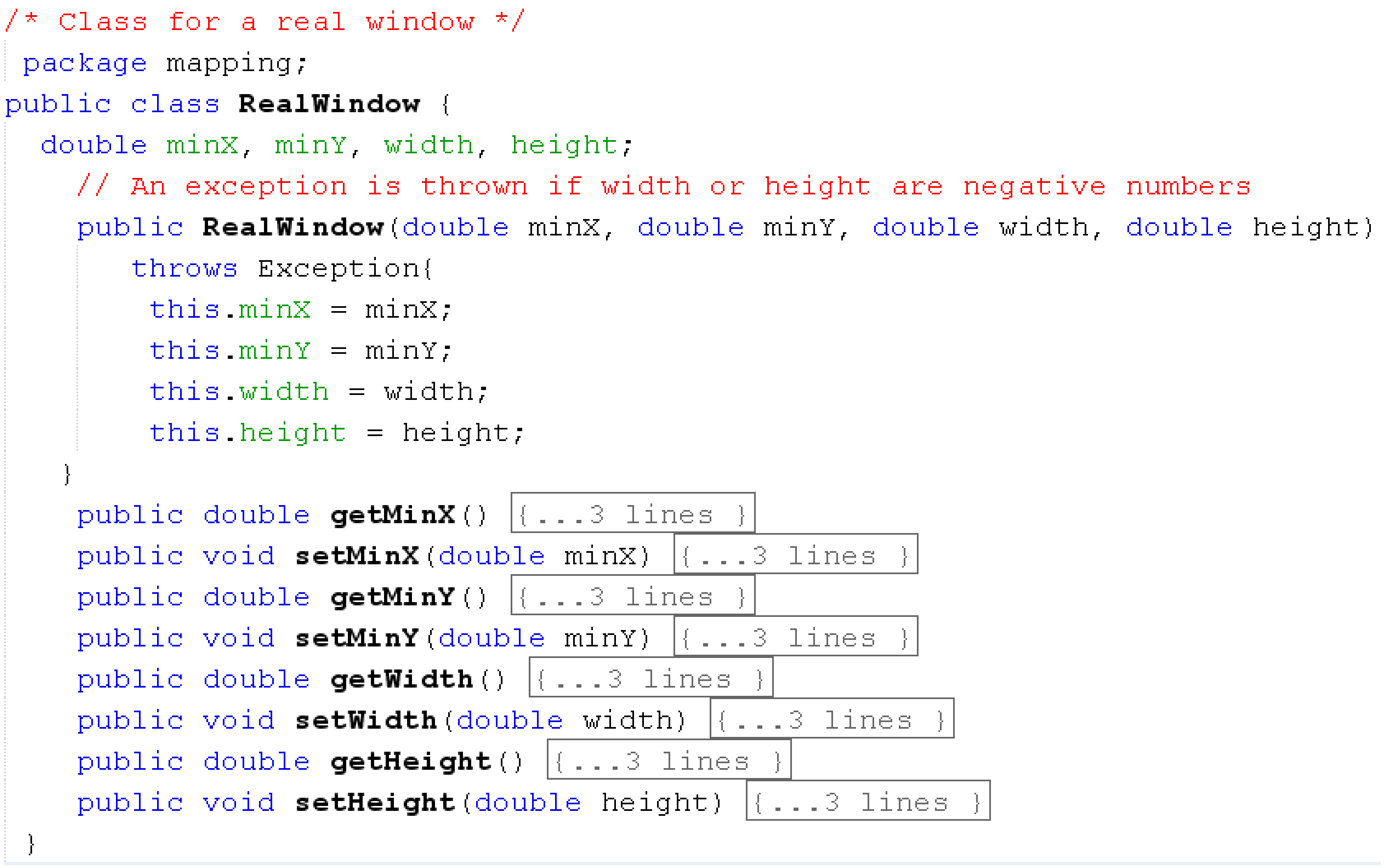


```java
/* Class encapsulates a list of device-points */
package mapping;
import java.util.ArrayList;
import java.awt.Point;

public class DevicePointList extends ArrayList<Point> {
    // Ctreate a list of device points with 0 elements
    public DevicePointList(){
        super();
    }
}
```

Demo.: Code

```java
/* Class for a device window */
 package mapping;

 public class DeviceWindow {
  public int left, top, width, height;
    // Exception is thrown if values are negative
    public DeviceWindow(int left, int top, int width, int height)
                                    throws Exception {
        this.left = left;
        this.top = top;
        this.width = width;
        this.height = height;
    }
    public int getLeft() {...3 lines }
    public void setLeft(int left) {...3 lines }
    public int getTop() {...3 lines }
    public void setTop(int top) {...3 lines }
    public int getWidth() {...3 lines }
    public void setWidth(int width) {...3 lines }
    public int getHeight() {...3 lines }
    public void setHeight(int height) {...3 lines }
}
```

Demo.: Code

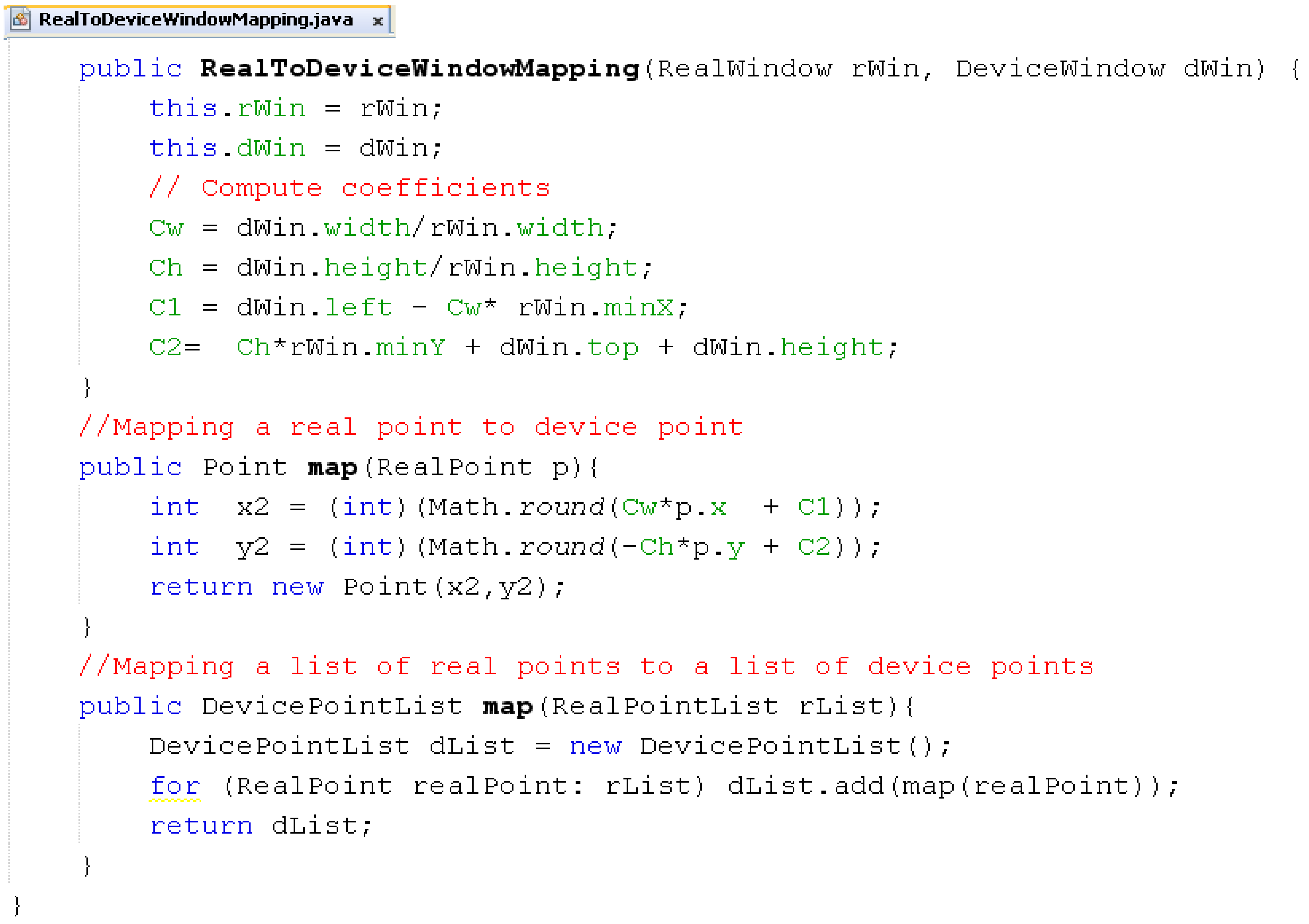


```
/* Class for a real window */
 package mapping;
public class RealWindow {
  double minX, minY, width, height;
    // An exception is thrown if width or height are negative numbers
    public RealWindow(double minX, double minY, double width, double height)
       throws Exception{
        this.minX = minX;
        this.minY = minY;
        this.width = width;
        this.height = height;
   }
    public double getMinX() {...3 lines }
    public void setMinX(double minX) {...3 lines }
    public double getMinY() {...3 lines }
    public void setMinY(double minY) {...3 lines }
    public double getWidth() {...3 lines }
    public void setWidth(double width) {...3 lines }
    public double getHeight() {...3 lines }
    public void setHeight(double height) {...3 lines }
 }
```

Demo.: Code

```
/* Class for mapping a real point to a point on a device
   Mapping a RealPointList to DevicPointList
*/
package mapping;
import java.awt.Point;
public class RealToDeviceWindowMapping {
    RealWindow rWin; // real window
    DeviceWindow dWin; // window on device
    /* Cw = dWin.width/rWin.width
       Ch = dWin.height/rWin.height
       C1 = dWin.left - Cw* rWin.minX
       C2=  Ch*rWin.minY + dWin.top + dWin.height
       x2 = Cwx1  + C1
       y2 = - Chy1 + C2
    */
    double Cw, Ch, C1, C2; // coefficients for mapping
```

Demo.: Code


RealToDeviceWindowMapping.java

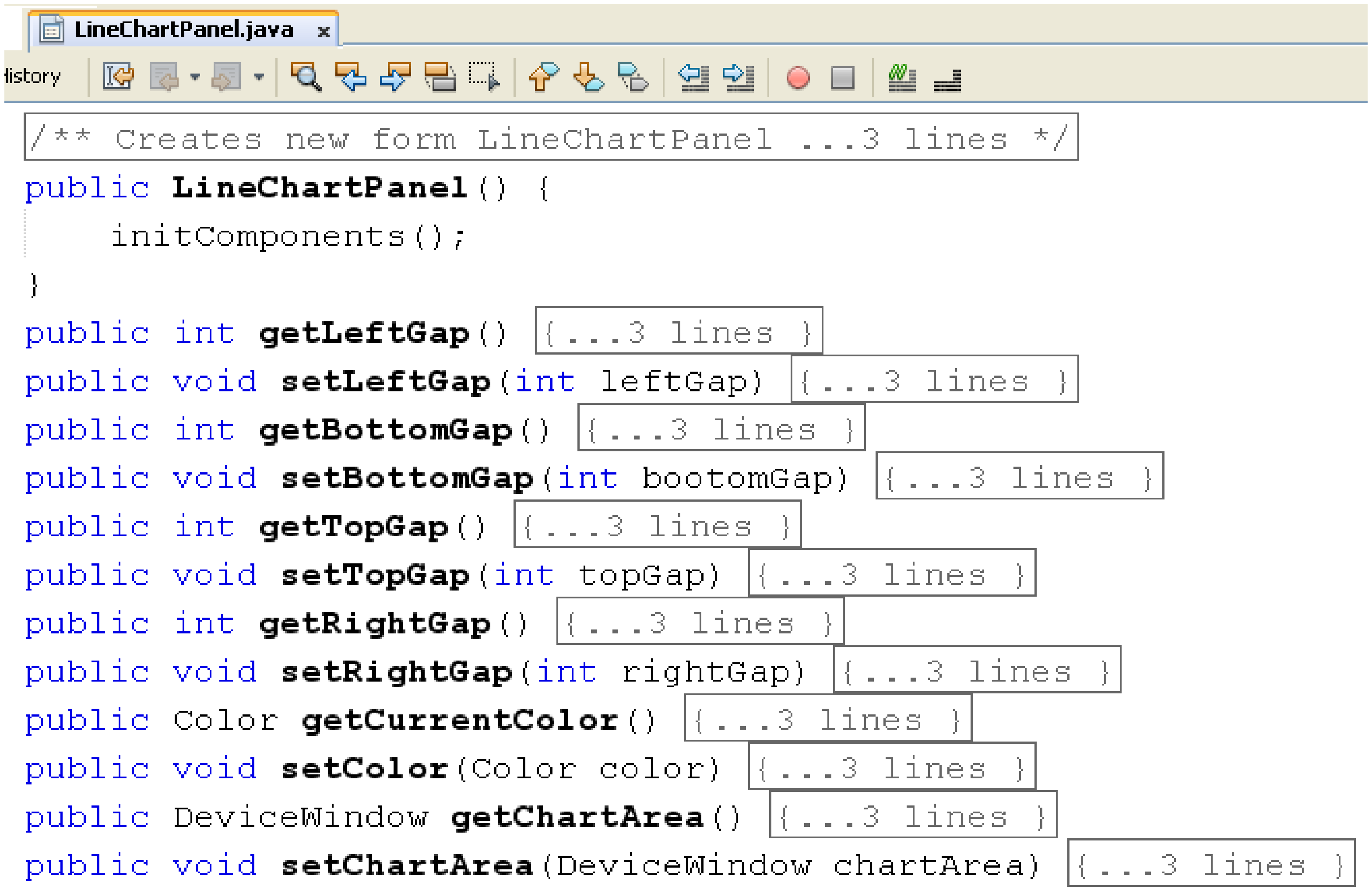
```java
    public RealToDeviceWindowMapping(RealWindow rWin, DeviceWindow dWin) {
        this.rWin = rWin;
        this.dWin = dWin;
        // Compute coefficients
        Cw = dWin.width/rWin.width;
        Ch = dWin.height/rWin.height;
        C1 = dWin.left - Cw* rWin.minX;
        C2=  Ch*rWin.minY + dWin.top + dWin.height;
    }
    //Mapping a real point to device point
    public Point map(RealPoint p){
        int  x2 = (int)(Math.round(Cw*p.x  + C1));
        int  y2 = (int)(Math.round(-Ch*p.y + C2));
        return new Point(x2,y2);
    }
    //Mapping a list of real points to a list of device points
    public DevicePointList map(RealPointList rList){
        DevicePointList dList = new DevicePointList();
        for (RealPoint realPoint: rList) dList.add(map(realPoint));
        return dList;
    }
}
```

Demo.: Code

```
/* Panel for drawing line charts */
package mapping;
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JOptionPane;
import java.awt.Point;
import java.awt.Font;
import java.awt.FontMetrics;
import java.awt.Graphics2D;
public class LineChartPanel extends javax.swing.JPanel {
    // gaps from boundary
    int leftGap=40; // gaps for presenting axes
    int topGap=10;
    int rightGap=10;
    int bottomGap=40;
    int innerGap=30; // gap between axes and chart area
    Color currentColor= Color.BLACK; // current coor is used
    DeviceWindow chartArea= null; // Drawing area
    int numberOfScale=4; // number of scale lines
```

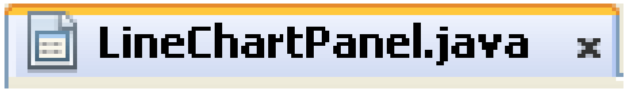
Demo.: Code

Constructor, getters, setters



```
/** Creates new form LineChartPanel ...3 lines */
public LineChartPanel() {
    initComponents();
}
public int getLeftGap() {...3 lines }
public void setLeftGap(int leftGap) {...3 lines }
public int getBottomGap() {...3 lines }
public void setBottomGap(int bootomGap) {...3 lines }
public int getTopGap() {...3 lines }
public void setTopGap(int topGap) {...3 lines }
public int getRightGap() {...3 lines }
public void setRightGap(int rightGap) {...3 lines }
public Color getCurrentColor() {...3 lines }
public void setColor(Color color) {...3 lines }
public DeviceWindow getChartArea() {...3 lines }
public void setChartArea(DeviceWindow chartArea) {...3 lines }
```

Demo.: Code

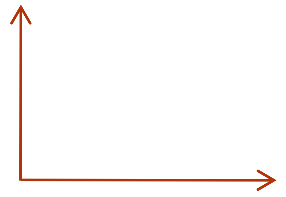

LineChartPanel.java

```
// From gaps, the chart area is computed
void setupChartArea(){
    int left = leftGap + innerGap;
    int top = topGap + innerGap;
    int width= this.getWidth() - leftGap - rightGap- 2*innerGap;
    int height = this.getHeight()- topGap - bottomGap- 2*innerGap;
    try {
        this.chartArea = new DeviceWindow(left, top, width, height);
    }
    catch (Exception e){
        String msg= "Device window parameters must be positine integers";
        JOptionPane.showMessageDialog(this, msg);
    }
}
```

Demo.: Code

LineChartPanel.java

```
void drawAxes(){
    Graphics g = this.getGraphics();
    int arrowLength=10, arrowWidth=2;
    // Draw horizontal axis
    int x1, y1, x2, y2;
    x1= x2=this.leftGap;
    y1= this.getHeight()-this.bottomGap;
    y2= this.topGap;
    g.drawLine(x1, y1, x2, y2);
    // Draw the arrow
    g.drawLine(x2-arrowWidth, y2+arrowLength, x2, y2);
    g.drawLine(x2+arrowWidth, y2+arrowLength, x2, y2);
    // Draw vetical axis
    x1= this.leftGap;
    x2= this.getWidth()-this.rightGap;
    y1= y2= this.getHeight()-this.bottomGap;
    g.drawLine(x1, y1, x2, y2);
    // Draw the arrow
    g.drawLine(x2, y2, x2-arrowLength, y2-arrowWidth);
    g.drawLine(x2, y2, x2-arrowLength, y2+arrowWidth);
}
```

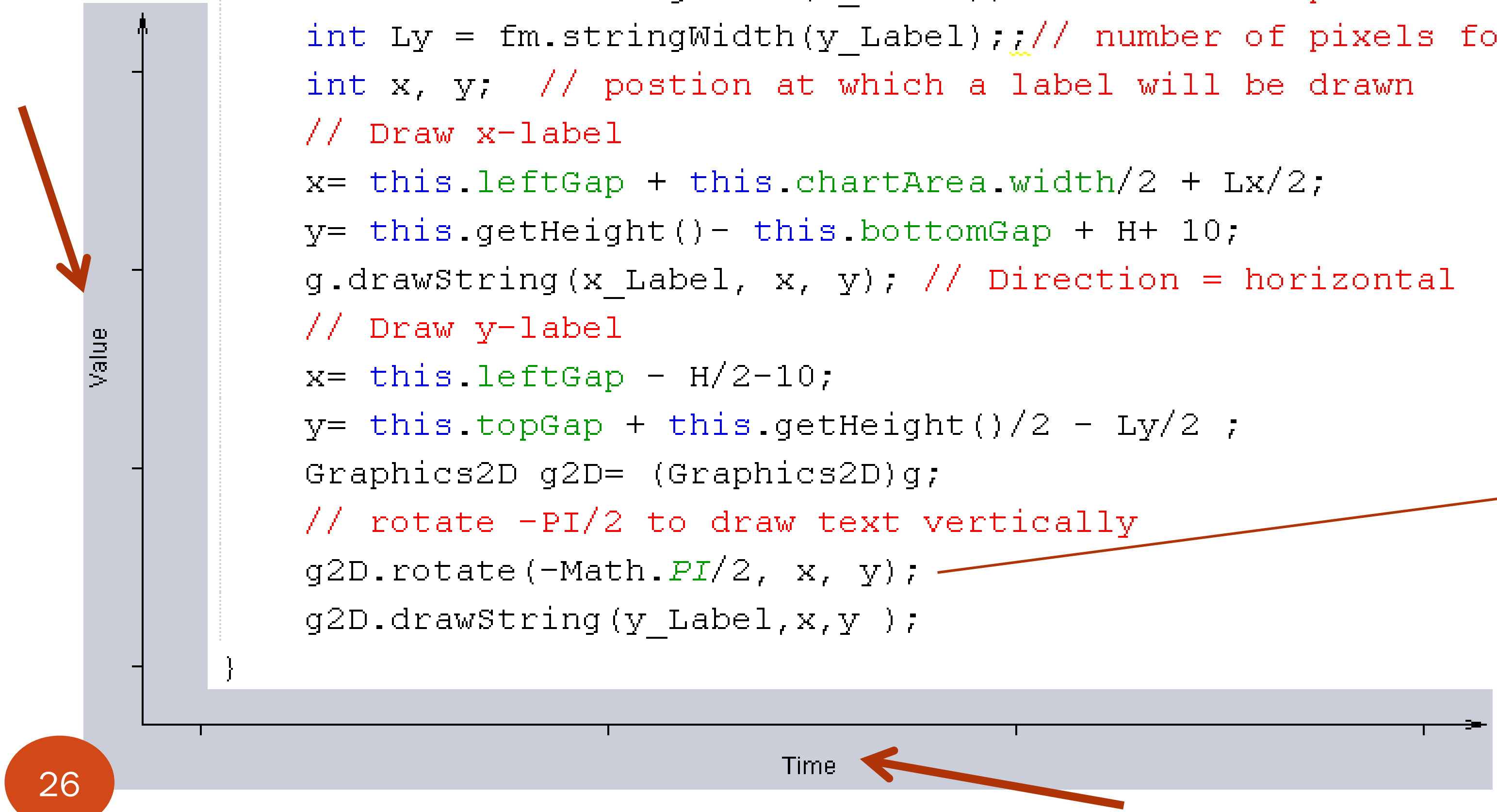


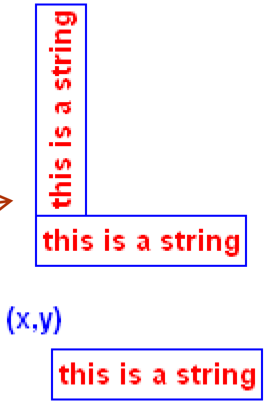
TRƯỜNG ĐẠI HỌC FPT

Demo.: Code

LineChartPanel.java

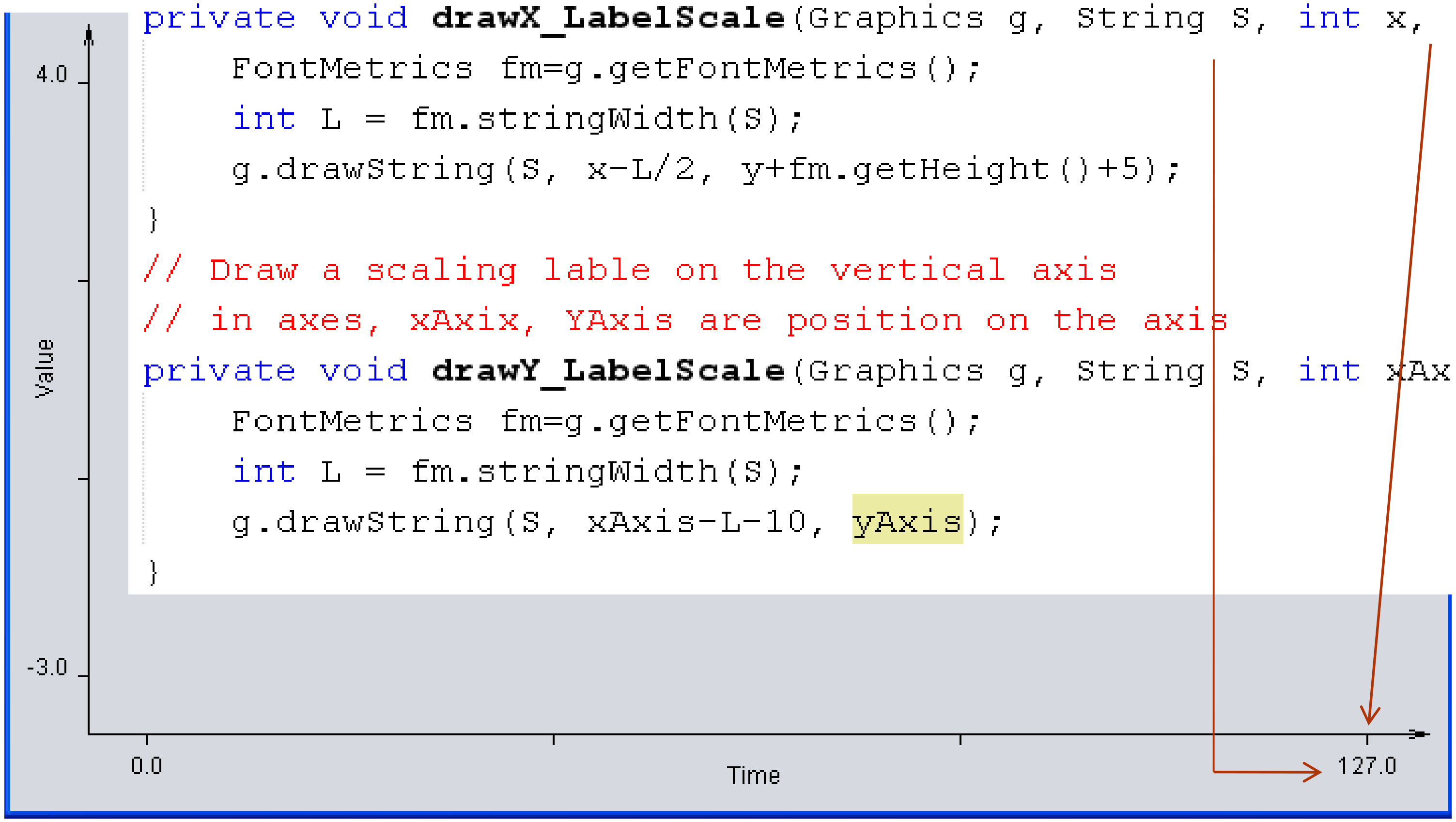
```
// Draw lables of axes
void drawLabels (String x_Label, String y_Label){
    Graphics g = this.getGraphics();
    Font font= g.getFont(); // Get font for computing positions of labels
    FontMetrics fm= g.getFontMetrics();
    int H = fm.getHeight();; // height of the current font
    int Lx = fm.stringWidth(x_Label); // number of pixels for x-label
    int Ly = fm.stringWidth(y_Label);;// number of pixels for y-label
    int x, y;  // postion at which a label will be drawn
    // Draw x-label
    x= this.leftGap + this.chartArea.width/2 + Lx/2;
    y= this.getHeight()- this.bottomGap + H+ 10;
    g.drawString(x_Label, x, y); // Direction = horizontal
    // Draw y-label
    x= this.leftGap - H/2-10;
    y= this.topGap + this.getHeight()/2 - Ly/2 ;
    Graphics2D g2D= (Graphics2D)g;
    // rotate -PI/2 to draw text vertically
    g2D.rotate(-Math.PI/2, x, y);
    g2D.drawString(y_Label,x,y );
}
```





Demo.: Code

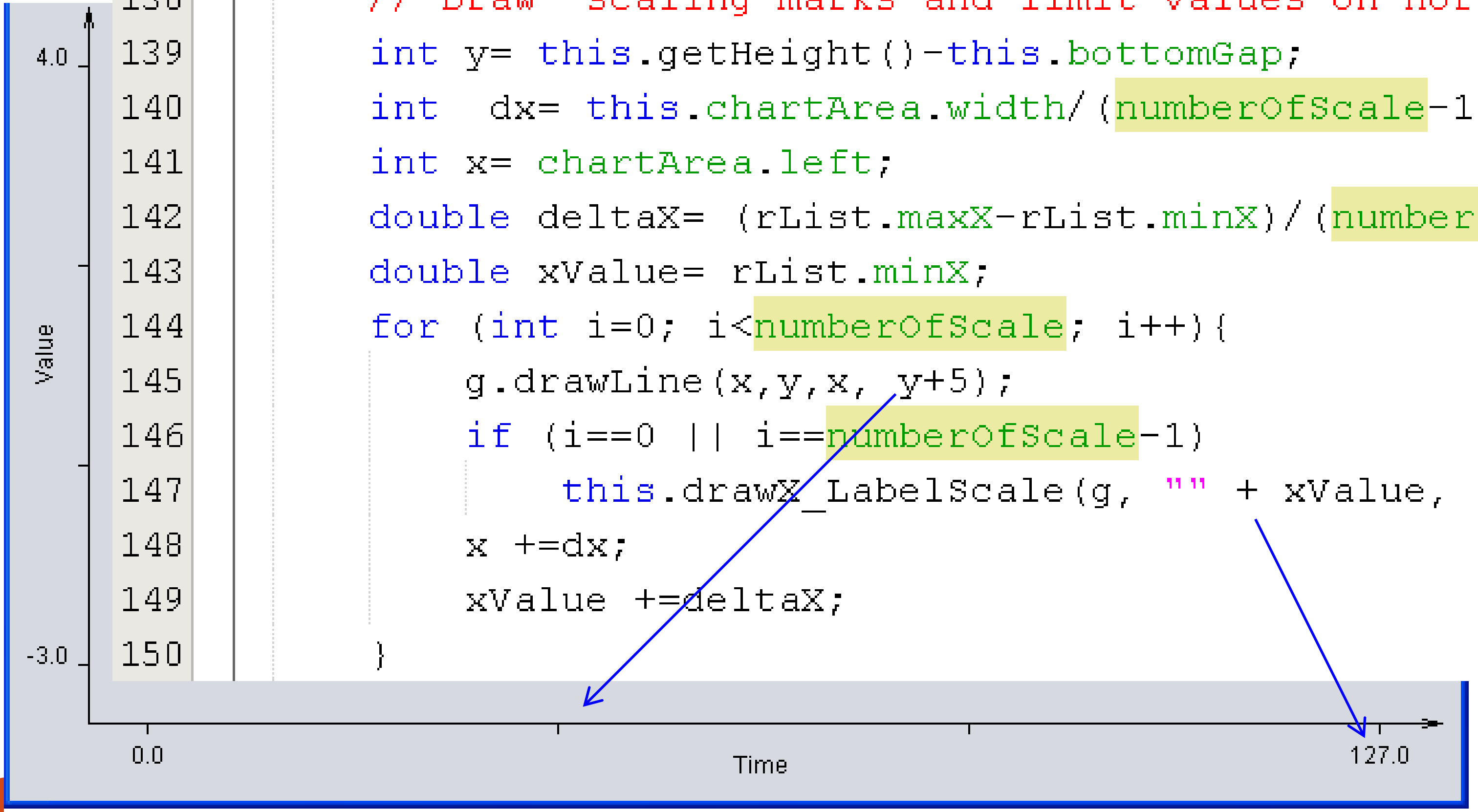
LineChartPanel.java



```java
// Draw scaling label on the horizontal axes
// (x,y): position on the horizontal axes
private void drawX_LabelScale(Graphics g, String S, int x, int y ){
    FontMetrics fm=g.getFontMetrics();
    int L = fm.stringWidth(S);
    g.drawString(S, x-L/2, y+fm.getHeight()+5);
}
// Draw a scaling lable on the vertical axis
// in axes, xAxix, YAxis are position on the axis
private void drawY_LabelScale(Graphics g, String S, int xAxis, int yAxis){
    FontMetrics fm=g.getFontMetrics();
    int L = fm.stringWidth(S);
    g.drawString(S, xAxis-L-10, yAxis);
}
```

Demo.: Code

LineChartPanel.java

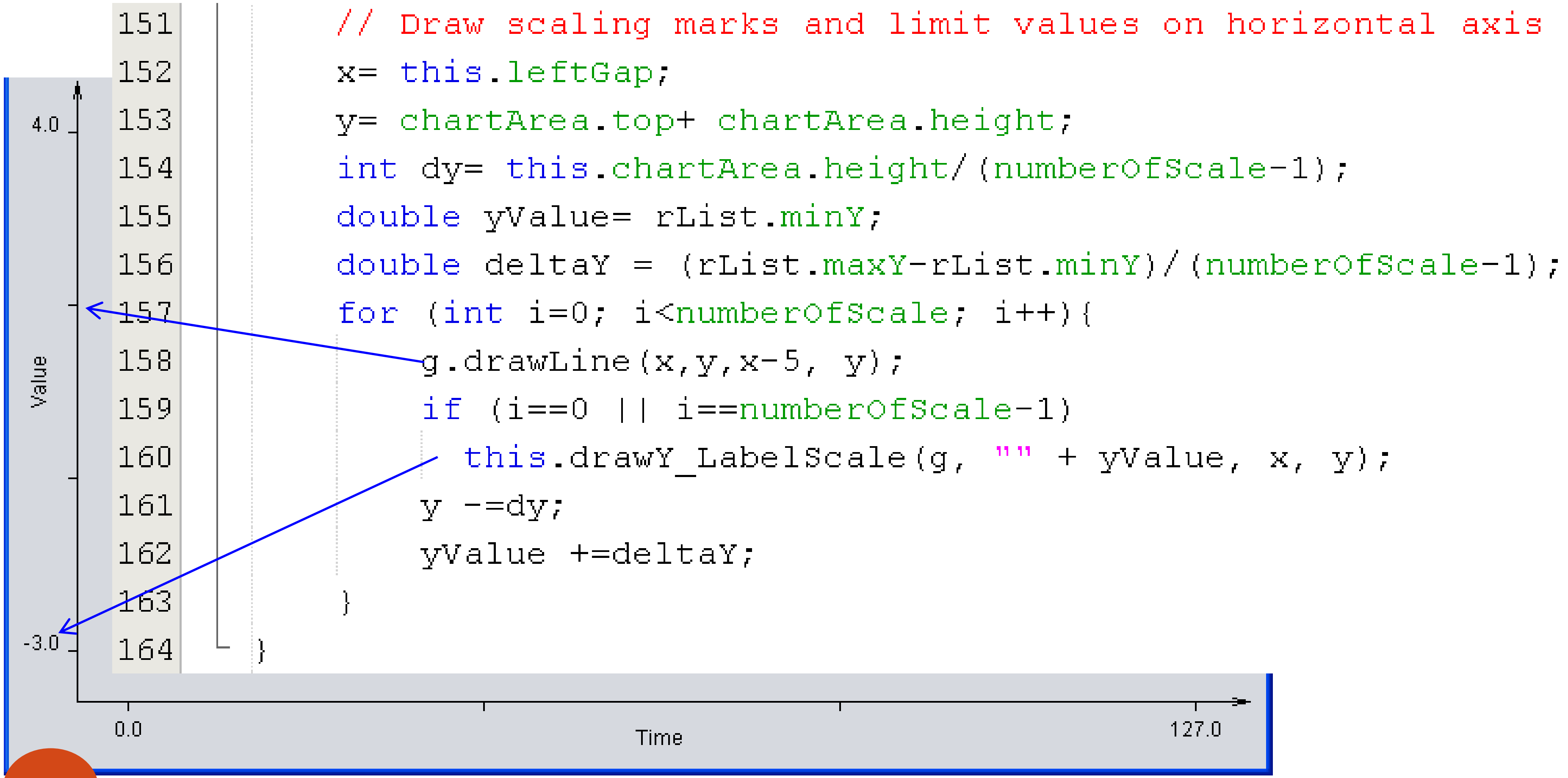


```
// Draw scaling marks and real limit values on axes
public void drawScale(RealPointList rList){
    Graphics g= this.getGraphics();
    // Draw  scaling marks and limit values on horizontal axis
    int y= this.getHeight()-this.bottomGap;
    int  dx= this.chartArea.width/(numberOfScale-1);
    int x= chartArea.left;
    double deltaX= (rList.maxX-rList.minX)/(numberOfScale-1);
    double xValue= rList.minX;
    for (int i=0; i<numberOfScale; i++){
        g.drawLine(x,y,x, y+5);
        if (i==0 || i==numberOfScale-1)
            this.drawX_LabelScale(g, "" + xValue, x, y);
        x +=dx;
        xValue +=deltaX;
    }
```

Demo.: Code

LineChartPanel.java

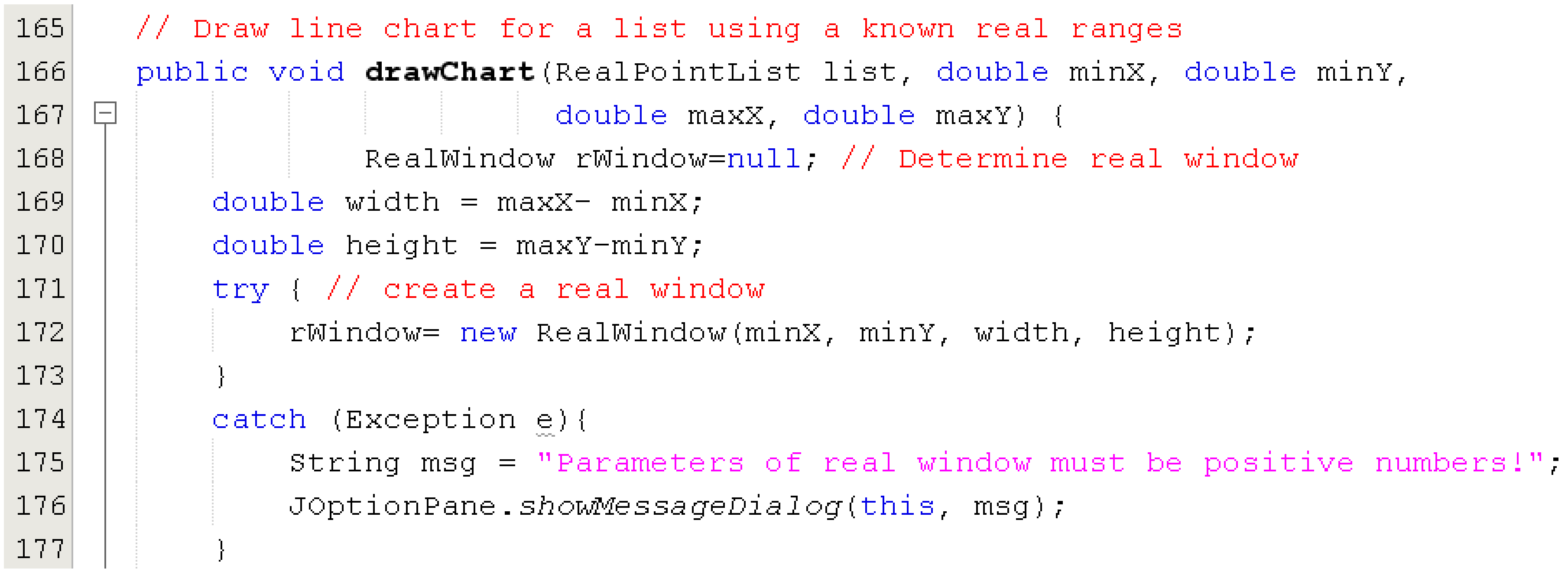
The method drawScale(..) contd.



```
        // Draw scaling marks and limit values on horizontal axis
        x= this.leftGap;
        y= chartArea.top+ chartArea.height;
        int dy= this.chartArea.height/(numberOfScale-1);
        double yValue= rList.minY;
        double deltaY = (rList.maxY-rList.minY)/(numberOfScale-1);
        for (int i=0; i<numberOfScale; i++){
            g.drawLine(x,y,x-5, y);
            if (i==0 || i==numberOfScale-1)
               this.drawY_LabelScale(g, "" + yValue, x, y);
            y -=dy;
            yValue +=deltaY;
        }
    }
```

Demo.: Code

LineChartPanel.java

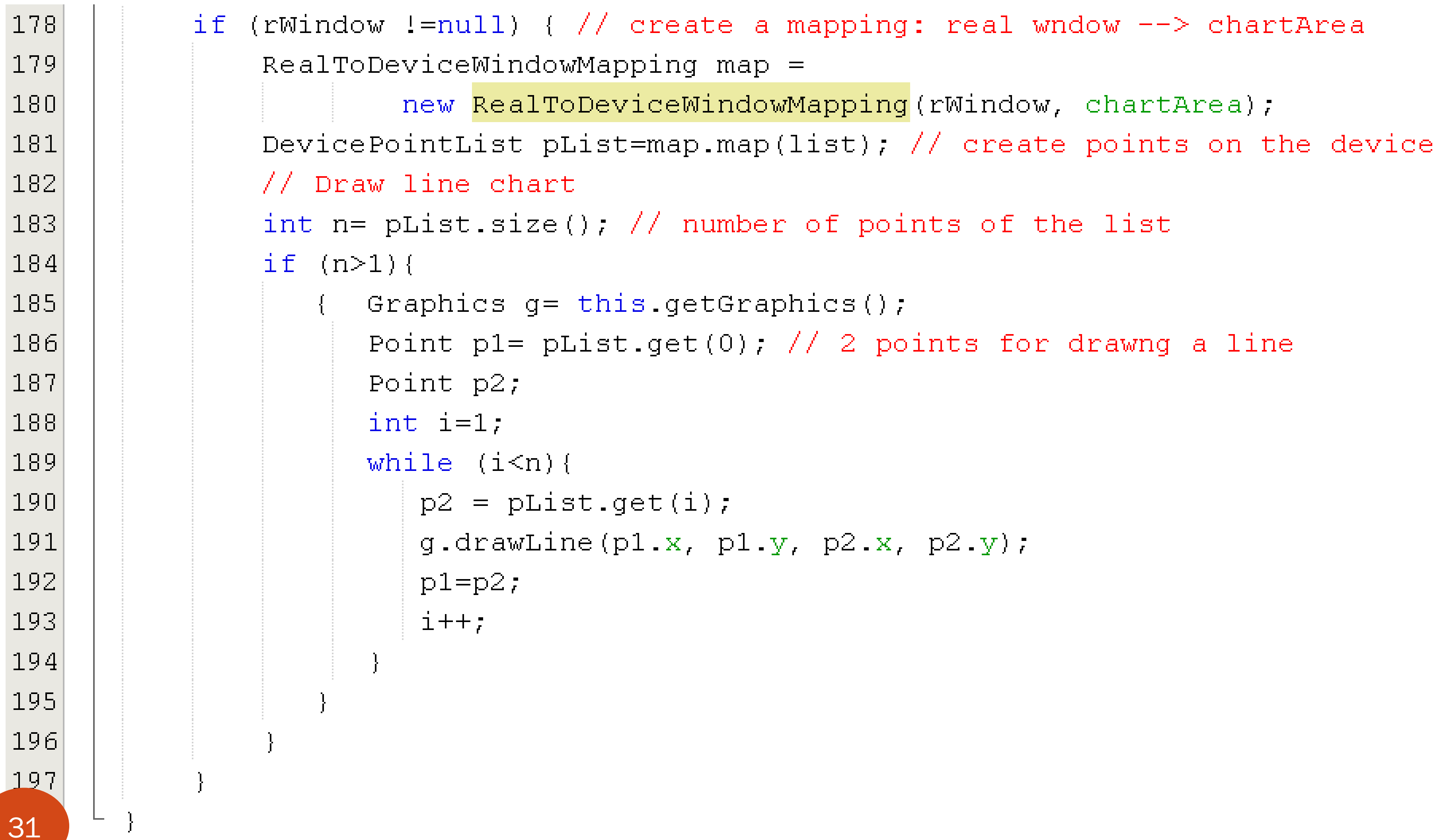


```java
// Draw line chart for a list using a known real ranges
public void drawChart(RealPointList list, double minX, double minY,
                      double maxX, double maxY) {
            RealWindow rWindow=null; // Determine real window
    double width = maxX- minX;
    double height = maxY-minY;
    try { // create a real window
        rWindow= new RealWindow(minX, minY, width, height);
    }
    catch (Exception e){
        String msg = "Parameters of real window must be positive numbers!";
        JOptionPane.showMessageDialog(this, msg);
    }
```

Demo.: Code

The method drawChart(...) contd.

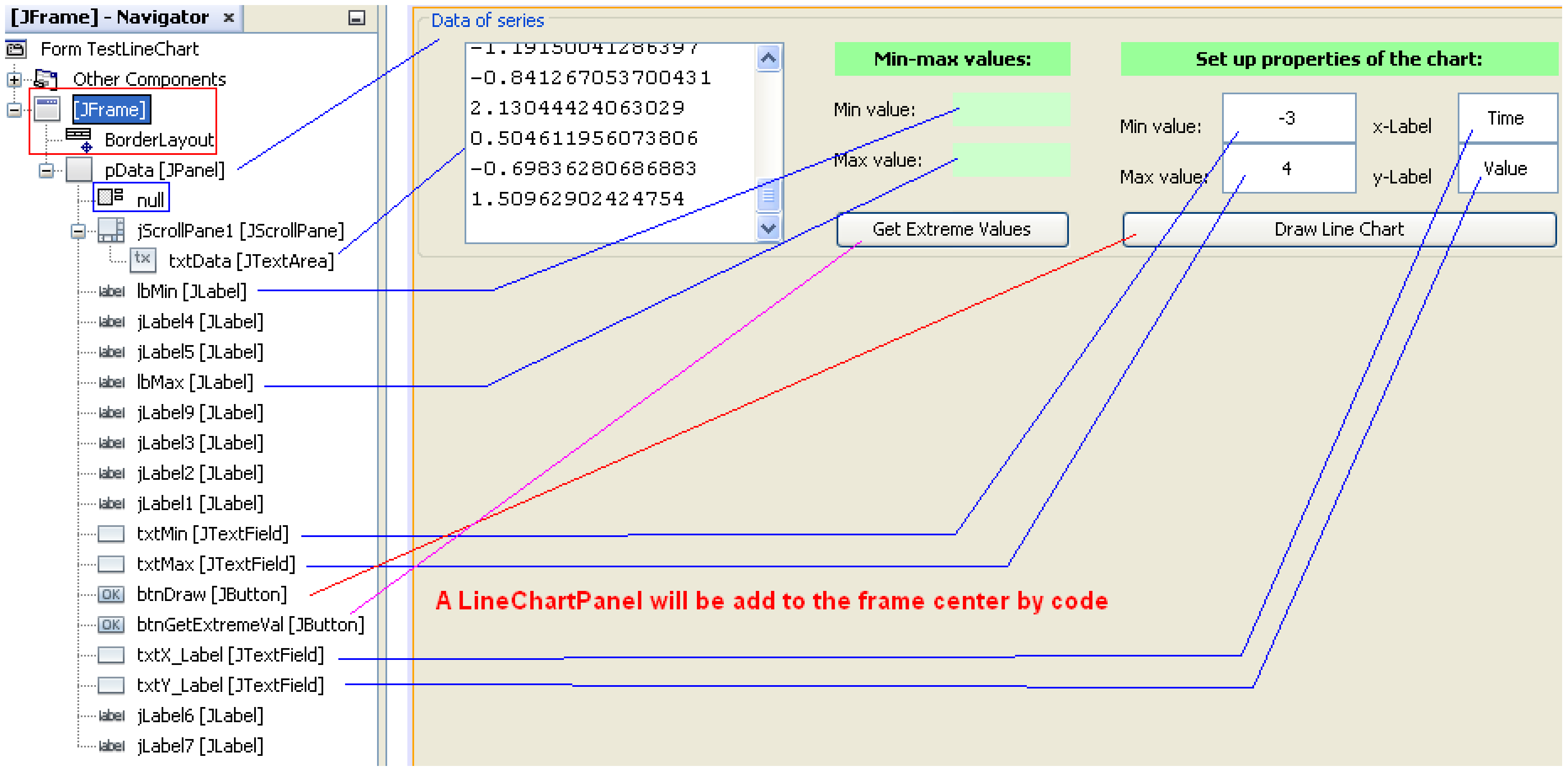
LineChartPanel.java



```java
        if (rWindow !=null) { // create a mapping: real wndow --> chartArea
            RealToDeviceWindowMapping map =
                    new RealToDeviceWindowMapping(rWindow, chartArea);
            DevicePointList pList=map.map(list); // create points on the device
            // Draw line chart
            int n= pList.size(); // number of points of the list
            if (n>1){
               {  Graphics g= this.getGraphics();
                  Point p1= pList.get(0); // 2 points for drawng a line
                  Point p2;
                  int i=1;
                  while (i<n){
                     p2 = pList.get(i);
                     g.drawLine(p1.x, p1.y, p2.x, p2.y);
                     p1=p2;
                     i++;
                  }
               }
            }
        }
    }
```

Demo.: Code

TestLineChart.java



```
[JFrame] - Navigator
Form TestLineChart
  Other Components
  [JFrame]
    BorderLayout
    pData [JPanel]
      null
      jScrollPane1 [JScrollPane]
        txtData [JTextArea]
      lbMin [JLabel]
      jLabel4 [JLabel]
      jLabel5 [JLabel]
      lbMax [JLabel]
      jLabel9 [JLabel]
      jLabel3 [JLabel]
      jLabel2 [JLabel]
      jLabel1 [JLabel]
      txtMin [JTextField]
      txtMax [JTextField]
      btnDraw [JButton]
      btnGetExtremeVal [JButton]
      txtX_Label [JTextField]
      txtY_Label [JTextField]
      jLabel6 [JLabel]
      jLabel7 [JLabel]
```

Data of series

```
-1.19150041286397
-0.841267053700431
2.13044424063029
0.504611956073806
-0.6983628068688з
1.50962902424754
```

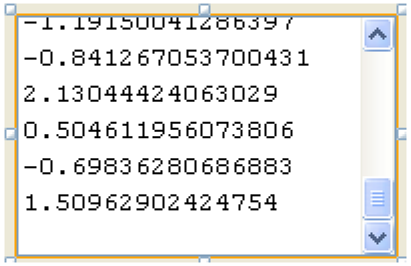
Min-max values:	
Min value:	
Max value:	

Set up properties of the chart:			
Min value:	-3	x-Label	Time
Max value:	4	y-Label	Value

A LineChartPanel will be add to the frame center by code

TRƯỜNG ĐẠI HỌC FPT

Demo.: Code

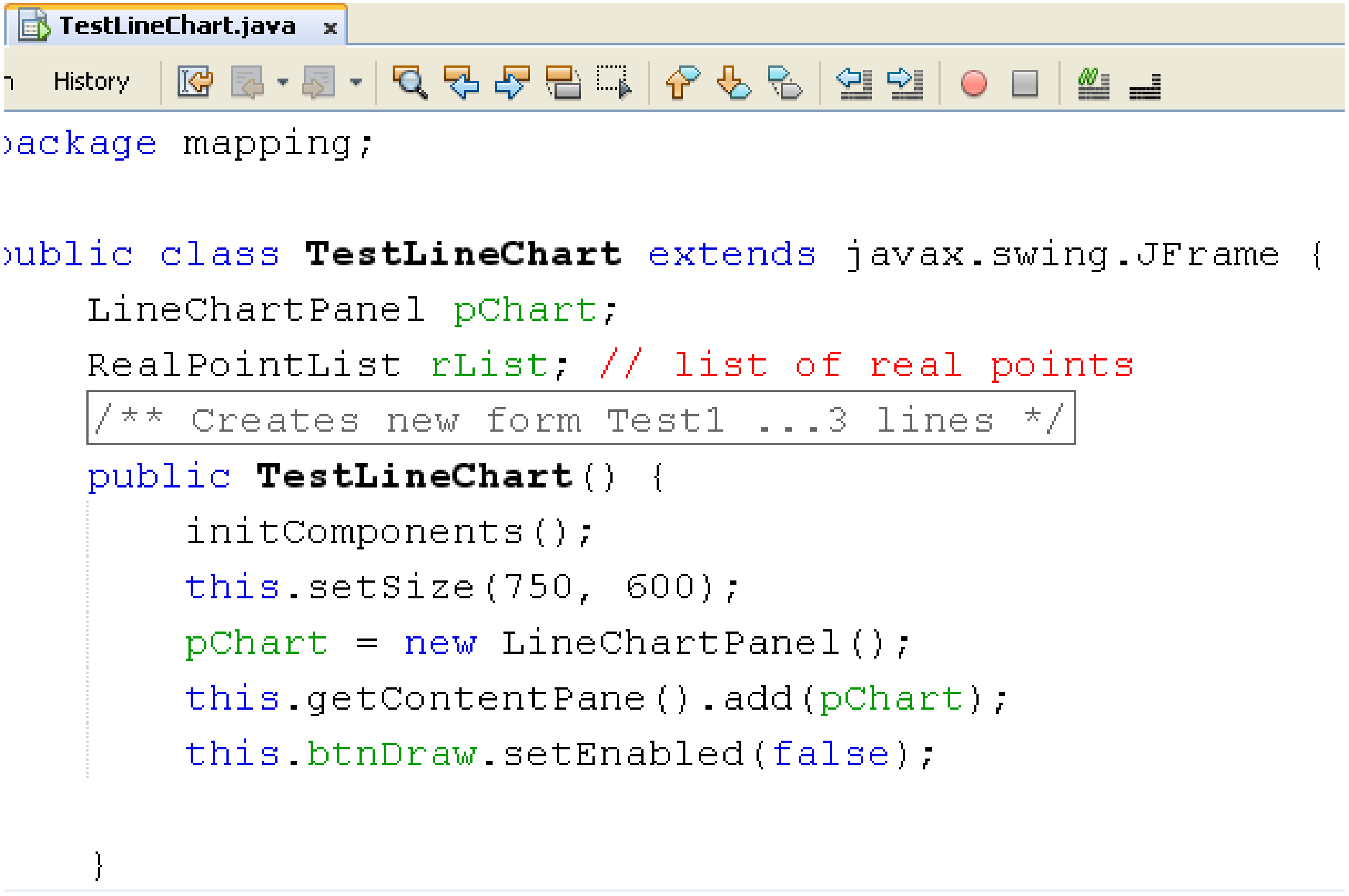


```
-1.19150041286397
-0.841267053700431
2.13044424063029
0.504611956073806
-0.698362806868830
1.50962902424754
```

Property Text:

```
-0.300104318204609
-0.530859690244707
-0.92209034994361
-0.906862571399915
-0.0166379969354826
-0.860008909883885
-0.175941438085159
-0.92209034994361
-0.191168727157551
0.0817553481397268
-0.300104318204609
-0.860008909883885
-0.0728626322272284
-0.191168727157551
-0.300104318204609
-0.00726706378588532
-0.0740340023176218
-0.0564639730644435
0.262143372601077
0.211775143973984
-0.0166379969354826
0.123924067712617
0.267999733581741
-0.0564639730644435
-0.191168727157551
-0.147829126965571
-0.273163491385384
-0.0728626322272284
0.318367962208834
0.262143372601077
0.211775143973984
-0.220452326788983
-0.184140571878031
0.0302161899464127
-0.281362462021154
-0.213424660980765
-0.295418935737296
0.0302161899464127
-0.22279509960119
-0.295418935737296
-0.213424660980765
-0.213424660980765
-0.0552926192897602
-0.220452326788983
0.494070049468728
0.494070049468728
0.511640111353326
0.426131758957036
-0.213424660980765
-0.295418935737296
0.053643037020138
0.0407579986572305
0.511640111353326
-0.418410511029192
-0.241536955784644
0.361707056613802
-0.447694110660624
0.426131758957036
0.257457990133764
-0.541402086321088
-0.327045308180934
-0.447694110660624
-0.903348493760155
-0.744045000400206
-0.744045000400206
-0.636281203652004
-0.453550961112591
-0.541402086321088
-0.418410511029192
-0.447694110660624
-0.879921744580691
-0.3680426087163
-0.541402086321088
-0.636281203652004
-0.636281203652004
-0.594112598289085
0.879443532401462
0.119238603666753
-0.57771384123204
-0.664393661612984
-0.57771384123204
-0.770985990376532
-0.761615551756108
-0.64916588306929
0.059499871156395
-0.3680426087163
0.523353812257261
-1.14816033951639
-0.761615551756108
0.119238603666753
0.603005803672886
-1.14816033951639
0.47650015074123
0.595977648393366
0.603005803672886
0.919268875480871
0.485870099890352
0.923954421105285
0.603005803672886
0.595977648393366
0.538581101329652
0.485870099890352
0.656887620468437
0.695542311348697
0.618233092745278
0.67328588805418
0.695542311348697
1.06803013592154
2.13044424063029
0.656887620468437
0.695542311348697
1.08794378640385
1.95825476259625
0.618233092745278
1.76849652793441
1.11722754919238
0.504611956073806
1.50962902424754
0.504611956073806
1.95825476259625
2.20423856580844
-1.19150041286397
-0.841267053700431
2.13044424063029
0.504611956073806
-0.69836280686883
1.50962902424754
```

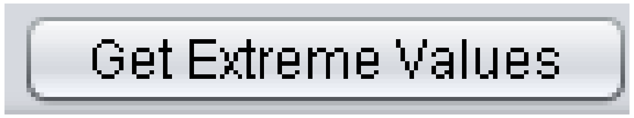
Demo.: Code

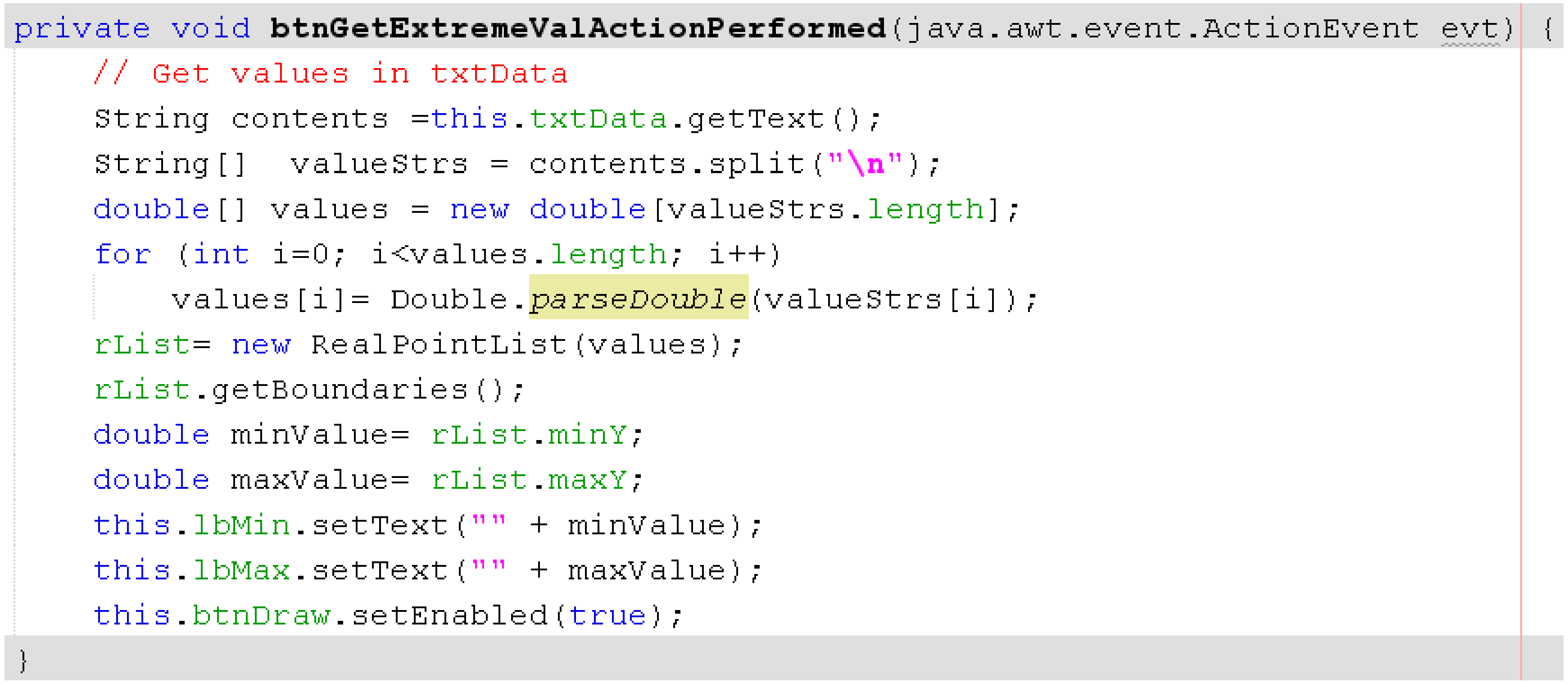


```
ackage mapping;

ublic class TestLineChart extends javax.swing.JFrame {
    LineChartPanel pChart;
    RealPointList rList; // list of real points
    /** Creates new form Test1 ...3 lines */
    public TestLineChart() {
        initComponents();
        this.setSize(750, 600);
        pChart = new LineChartPanel();
        this.getContentPane().add(pChart);
        this.btnDraw.setEnabled(false);

    }
```

TRƯỜNG ĐẠI HỌC FPT

Demo.: Code

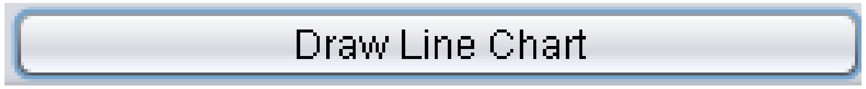




```
private void btnGetExtremeValActionPerformed(java.awt.event.ActionEvent evt) {
    // Get values in txtData
    String contents =this.txtData.getText();
    String[]  valueStrs = contents.split("\n");
    double[] values = new double[valueStrs.length];
    for (int i=0; i<values.length; i++)
        values[i]= Double.parseDouble(valueStrs[i]);
    rList= new RealPointList(values);
    rList.getBoundaries();
    double minValue= rList.minY;
    double maxValue= rList.maxY;
    this.lbMin.setText("" + minValue);
    this.lbMax.setText("" + maxValue);
    this.btnDraw.setEnabled(true);
}
```

TRƯỜNG ĐẠI HỌC FPT

Demo.: Code



```
private void btnDrawActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    // Get setup vaules
    double minY = Double.parseDouble(txtMin.getText());
    double maxY = Double.parseDouble(txtMax.getText());
    String x_Label= txtX_Label.getText();
    String y_Label= txtY_Label.getText();
    if (rList!=null) {
        //Set up min value, max value to real point list
        rList.minY= minY;
        rList.maxY= maxY;
        rList.minX=0; // x= 0, 1, 2, ...
        rList.maxX = rList.size();
        // compute Chart area
        this.pChart.setupChartArea();
        this.pChart.drawAxes();
        this.pChart.drawLabels(x_Label, y_Label);
        this.pChart.drawScale(rList);
        double minX= 0, maxX= rList.size();
        pChart.drawChart(rList, minX, minY, maxX, maxY);
    }
}
```

Summary

- Normal Coordinates versus Device Coordinates
- Expected Shape
- Mapping a Real Point to Device Point
- Drawing Area in a Component

Thank You