Instructions

1. Main Function
   1. Draw the chart according the data by user providing.
2. Function Instructions
   1. DataPoint.cs
      1. Property
         1. index: int

//Mark the value’s column number.

* + - 1. value: object

//The data value.

* + - 1. columnName: string

//The name of value’s column.

* + 1. Method
       1. DataPoint():

//To initialize the DataPoint class.

* 1. DataSeries.cs
     1. Property
        1. dataPoints: List<DataPoint>

//A column data.

* + - 1. rowName: string

//The name of row.

* + - 1. index: int

//Mark the value’s row number.

* + 1. Method
       1. DataSeries():

//To initialize the DataSeries class.

* + - 1. ReadColumnsName(): List<string>

//To read the name of columns.

* 1. ChartData.cs
     1. Property
        1. dataSeries: List<DataSeries>

//A series of data.

* + - 1. title: string

//Data title.

* + - 1. scaling: float

//The scaling ratio between the data and image size.

* + 1. Method
       1. ChartData():

//To initialize the ChartData class.

* + - 1. ReadRowsName():List<string>

//To Read the name of rows.

* + - 1. CalculateColumnsCount():int

//To calculate the columns count.

* + - 1. AddRowData(DataSeries newDataSeries): void

//To add a new row data.

//newDataSeries: a new row data.

* + - 1. AddColumnData(DataPoint[] dataPoints): void

//To add new column data.

//dataPoints: new column data.

* + - 1. Transpose():void

//To exchange data.

* 1. DataProvider.cs
     1. Property
        1. path: string

//Data source.

* + 1. Method
       1. DataProvider():

//To initialize the DataProvider class.

* + - 1. DataProvider(string path):

//To initialize the DataProvider class.

* + - 1. Load():ChartData

//To load data form data source.

* + - 1. LoadCore():ChartData

//To load data form data source.

* 1. ExcelDataProvider.cs
     1. Method
        1. ExcelDataProvider():

//To initialize the DataProvider class.

* + - 1. ExcelDataProvider(string path):

//To initialize the DataProvider class.

* + - 1. LoadCore():ChartData

//To load data form data source.

* 1. ShapeStyle.cs
     1. Property
        1. lineColor: Color

//Line color

* + - 1. fillColor: Color

//Fill color

* + 1. Method
       1. ShapeStyle():

//To initialize the ShapeStyle class.

* + - 1. ShapeStyle(Color lineColor, Color fillColor):

//To initialize the ShapeStyle class.

//lineColor: line color.

//fillColor: fill color.

* 1. TextStyle.cs
     1. Property
        1. textColor: Color

//Text color

* + 1. Method
       1. TextStyle():

//To initialize the TextStyle class.

* + - 1. TextStyle(Color color):

//To initialize the TextStyle class.

// color: text color.

* 1. ChartList.cs
     1. Property
        1. charts: List<Chart>

//A list of chart.

* + - 1. dataList: List<ChartData>

//A list of data.

* + 1. Method
       1. ChartList():

//To initialize the ChartList class.

* + - 1. Draw(Painter painter, Orientation orientation): void

//Draw image.

//painter: an object of Painter class.

//orientation: an object of enum.

* + - 1. Refresh(Painter painter, Orientation orientation): void

//Refresh image.

//painter: an object of Painter class.

//orientation: an object of enum.

* 1. Chart.cs
     1. Property
        1. data: ChartData

//Chart data.

* + - 1. textStyle: TextStyle

//Text style.

* + - 1. shapeStyle: ShapeStyle[]

//Shape style.

* + - 1. orientation: Orientation

//To choose the data row or column as image’s row or column.

* + - 1. chartType: ChartType

//Chart type.

* + - 1. xMax: float
      2. xMin: float
      3. yMax: float
      4. yMin: float

//The image border.

* + 1. Method
       1. Chart():

//To initialize the Chart class.

* + - 1. Chart(ChartData data):

//To initialize the Chart class.

//data: chart data.

* + - 1. Draw(Painter painter, Orientation orientation): void

//Draw image.

//painter: an object of Painter class.

//orientation: an object of enum.

* + - 1. DrawCore(Painter painter): void

//Draw image.

//painter: an object of Painter class.

* + - 1. Refresh(Painter painter): void

//Refresh image.

//painter: an object of Painter class.

* + - 1. RefreshCore(Painter painter): void

//Refresh image.

//painter: an object of Painter class.

* + - 1. DrawFrame(Painter painter): void

//Draw chart frame.

//painter: an object of Painter class.

* + - 1. DrawLegend(Painter painter): void

//Draw chart legend.

//painter: an object of Painter class.

* + - 1. DrawTitle(Painter painter): void

//Draw chart title.

//painter: an object of Painter class.

* 1. ColumnChart.cs
     1. Method
        1. ColumnChart():

//To initialize the ColumnChart class.

* + - 1. ColumnChart(ChartData data):

//To initialize the ColumnChart class.

//data: chart data.

* + - 1. DrawCore(Painter painter): void

//Draw column chart.

//painter: an object of Painter class.

* + - 1. RefreshCore(Painter painter): void

//Refresh column chart.

//painter: an object of Painter class.

* + - 1. DrawColumns(Painter painter): void

//Draw column.

//painter: an object of Painter class.

* + - 1. DrawLabel(Painter painter): void

//Draw chart label.

//painter: an object of Painter class.

* 1. LineChart.cs
     1. Method
        1. LineChart():

//To initialize the LineChart class.

* + - 1. LineChart(ChartData data):

//To initialize the LineChart class.

//data: chart data.

* + - 1. DrawCore(Painter painter): void

//Draw line chart.

//painter: an object of Painter class.

* + - 1. RefreshCore(Painter painter): void

//Refresh line chart.

//painter: an object of Painter class.

* + - 1. DrawLines(Painter painter): void

//Draw lines.

//painter: an object of Painter class.

* + - 1. DrawLabel(Painter painter): void

//Draw chart label.

//painter: an object of Painter class.

* 1. CurveChart.cs
     1. Method
        1. CurveChart():

//To initialize the CurveChart class.

* + - 1. CurveChart(ChartData data):

//To initialize the CurveChart class.

//data: chart data.

* + - 1. DrawCore(Painter painter): void

//Draw curve chart.

//painter: an object of Painter class.

* + - 1. RefreshCore(Painter painter): void

//Refresh curve chart.

//painter: an object of Painter class.

* + - 1. DrawCurve(Painter painter): void

//Draw curve.

//painter: an object of Painter class.

* + - 1. DrawLabel(Painter painter): void

//Draw chart label.

//painter: an object of Painter class.

* 1. PieChart.cs
     1. Method
        1. PieChart():

//To initialize the PieChart class.

* + - 1. PieChart(ChartData data):

//To initialize the PieChart class.

//data: chart data.

* + - 1. DrawCore(Painter painter): void

//Draw pie chart.

//painter: an object of Painter class.

* + - 1. RefreshCore(Painter painter): void

//Refresh pie chart.

//painter: an object of Painter class.

* 1. ChartPoint.cs
     1. Property
        1. x: float

//x-coordinate.

* + - 1. y: float

//y-coordinate.

* + 1. Method
       1. ChartPoint():

//To initialize the ChartPoint class.

* + - 1. ChartPoint(float x, float y):

//To initialize the ChartPoint class.

//x: x-coordinate.

//y: y-coordinate.

* 1. ChartMap.cs
     1. Property
        1. map: object

//map: image data.

* + - 1. width: float

//width: image width.

* + - 1. height: float

//height: image height.

* + 1. Method
       1. ChartMap():

//To initialize the ChartMap class.

* + - 1. ChartMap(Image map):

//To initialize the ChartMap class.

//map: image data.

* 1. Painter.cs
     1. Property
        1. chartMap: ChartMap

//chartMap: chart image.

* + 1. Method
       1. Painter():

//To initialize the Painter class.

* + - 1. DrawString(string s, TextStyle style, int textWidth, float x, float y): void

//Draw text on chart.

//s: text.

//style: text style.

//textWidth: text width.

//x: x-coordinate.

//y: y-coordinate.

* + - 1. DrawStringCore(string s, TextStyle style, int textWidth, float x, float y): void

//Draw text on chart.

//s: text.

//style: text style.

//textWidth: text width.

//x: x-coordinate.

//y: y-coordinate.

* + - 1. DrawLine(ShapeStyle style, IEnumerable<ChartPoint> points, int lineWidth, float tension): void

//Draw line on chart.

//style: shape style.

//points: chart points.

//lineWidth: line width.

//tension: the tension of the curve.

* + - 1. DrawLineCore(ShapeStyle style, IEnumerable<ChartPoint> points, int lineWidth, float tension): void

//Draw line on chart.

//style: shape style.

//points: chart points.

//lineWidth: line width.

//tension: the tension of the curve.

* + - 1. DrawPolygon(ShapeStyle style, IEnumerable<ChartPoint> points): void

//Draw polygon on chart.

//style: shape style.

//points: chart points.

* + - 1. DrawPolygonCore(ShapeStyle style, IEnumerable<ChartPoint> points): void

//Draw polygon on chart.

//style: shape style.

//points: chart points.

* + - 1. DrawRectangle(ShapeStyle style, ChartPoint point, float width, float height): void

//Draw rectangle on chart.

//style: shape style.

//point: chart point.

//width: rectangle width.

//height: rectangle height.

* + - 1. DrawRectangleCore(ShapeStyle style, ChartPoint point, float width, float height): void

//Draw rectangle on chart.

//style: shape style.

//point: chart point.

//width: rectangle width.

//height: rectangle height.

* + - 1. DrawCircular(ShapeStyle style, float x, float y, float width, float height, float startAngle, float sweepAngle): void

//Draw circular on chart.

//style: shape style.

//x: x-coordinate.

//y: y-coordinate.

//width: circular width.

//height: circular height.

//startAngle: pie startAngle.

//sweepAngle: pie sweepAngle.

* + - 1. DrawCircularCore(ShapeStyle style, float x, float y, float width, float height, float startAngle, float sweepAngle): void

//Draw circular on chart.

//style: shape style.

//x: x-coordinate.

//y: y-coordinate.

//width: circular width.

//height: circular height.

//startAngle: pie startAngle.

//sweepAngle: pie sweepAngle.

* + - 1. DrawImage(ChartMap chartMap, int x, int y): void

//Draw image on chart.

//chartMap: chart image.

//x: x-coordinate.

//y: y-coordinate.

* + - 1. DrawImageCore(ChartMap chartMap, int x, int y): void

//Draw image on chart.

//chartMap: chart image.

//x: x-coordinate.

//y: y-coordinate.

* + - 1. DrawImage(ChartMap chartMap, int x, int y, int width, int height): void

//Draw image on chart.

//chartMap: chart image.

//x: x-coordinate.

//y: y-coordinate.

//width: image width.

//height: image height.

* + - 1. DrawImageCore(ChartMap chartMap, int x, int y, int width, int height): void

//Draw image on chart.

//chartMap: chart image.

//x: x-coordinate.

//y: y-coordinate.

//width: image width.

//height: image height.

* + - 1. DrawImage(ChartMap chartMap, IEnumerable<ChartPoint> points): void

//Draw image on chart.

//chartMap: chart image.

//points: chart points.

* + - 1. DrawImageCore(ChartMap chartMap, IEnumerable<ChartPoint> points): void

//Draw image on chart.

//chartMap: chart image.

//points: chart points.

* 1. GdiPlusPainter.cs
     1. Property
        1. graphics: Graphics

//Draw on graphics.

* + 1. Method
       1. GdiPlusPainter():

//To initialize the GdiPlusPainter class.

* + - 1. GdiPlusPainter(Image image):

//To initialize the GdiPlusPainter class.

//image: image data.

* + - 1. DrawStringCore(string s, TextStyle style, int textWidth, float x, float y): void

//Draw text on chart.

//s: text.

//style: text style.

//textWidth: text width.

//x: x-coordinate.

//y: y-coordinate.

* + - 1. DrawLineCore(ShapeStyle style, IEnumerable<ChartPoint> points, int lineWidth, float tension): void

//Draw line on chart.

//style: shape style.

//points: chart points.

//lineWidth: line width.

//tension: the tension of the curve.

* + - 1. DrawPolygonCore(ShapeStyle style, IEnumerable<ChartPoint> points): void

//Draw polygon on chart.

//style: shape style.

//points: chart points.

* + - 1. DrawRectangleCore(ShapeStyle style, ChartPoint point, float width, float height): void

//Draw rectangle on chart.

//style: shape style.

//point: chart point.

//width: rectangle width.

//height: rectangle height.

* + - 1. DrawCircularCore(ShapeStyle style, float x, float y, float width, float height, float startAngle, float sweepAngle): void

//Draw circular on chart.

//style: shape style.

//x: x-coordinate.

//y: y-coordinate.

//width: circular width.

//height: circular height.

//startAngle: pie startAngle.

//sweepAngle: pie sweepAngle.

* + - 1. DrawImageCore(ChartMap chartMap, int x, int y): void

//Draw image on chart.

//chartMap: chart image.

//x: x-coordinate.

//y: y-coordinate.

* + - 1. DrawImageCore(ChartMap chartMap, int x, int y, int width, int height): void

//Draw image on chart.

//chartMap: chart image.

//x: x-coordinate.

//y: y-coordinate.

//width: image width.

//height: image height.

* + - 1. DrawImageCore(ChartMap chartMap, IEnumerable<ChartPoint> points): void

//Draw image on chart.

//chartMap: chart image.

//points: chart points.

* 1. BasicArithmetic.cs
     1. Method
        1. CalculateDistance(ChartData data, float xMax, float xMin): float

//Calculate the distance of x-coordinate.

//data: chart data.

//xMax: the x-max of the chart.

//xMin: the x-min of the chart.

* + - 1. CalculateRanks(ChartData data, out int rowsCount, out int columnsCount): void

//Calculate the chart border.

//data: chart data.

//rowsCount: the count of rows.

//columnsCount: the count of columns.

* + - 1. CalculateDataRatio(ChartData data): float[,]

//Calculate the data ratio.

//data: chart data.

* + - 1. CalculateMax(ChartData data): float

//Calculate the max value of data.

//data: chart data;

* + - 1. CalculateScaling(ref ChartData data, float yMax): void

//Calculate scaling ratio between the data and image size.

//data: chart data.

//yMax: the y-max of chart.

* 1. ChartView.cs
     1. Property
        1. dataPath: string

//Data source.

* + - 1. data: ChartData

//Chart data.

* + - 1. chartMap: ChartMap

//Chart image.

* + - 1. orientation: Orientation

//To choose the data row or column as image’s row or column.

* + - 1. chart: Chart

//An object of Chart class.

* + - 1. painter: Painter

//An object of Painter class.

* + - 1. chartType: ChartType

//Chart Type.

* + 1. Method
       1. ChartView():

//To initialize the ChartView class.

* + - 1. OnPaint(): void

//To draw chart.

* + - 1. Transpose(): void

//To exchange data.

* + - 1. SaveImage(string path): void

//To save chart into image.

1. Mainly using the process
   1. Add reference to your item.
   2. Add the control to your toolbox.
   3. Provide a path of data source.
   4. Read data form data source by DataProvider class.
   5. Create an object of chart class by data.
   6. Create an object of bitmap class.
   7. Create an object of painter class by bitmap.
   8. Draw chart.