# 鸿蒙开源第三方组件——MPAndroidChart图表绘制组件

## 前言

本组件是基于安卓平台的图片裁剪组件MPAndroidChart（  
https://github.com/PhilJay/MPAndroidChart），实现了其核心功能的鸿蒙化迁移和重构。目前代码已经开源到（https://gitee.com/isrc\_ohos/mpandroid-chart\_ohos），欢迎各位下载使用并提出宝贵意见！

## 背景

安卓版本的MPAndroidChart在GitHub上有超过3.3万个Star和8.3k个Fork，应该说是目前使用最广，体验最佳的开源图表库。它具绘制折线图、饼图、雷达图等图表的能力，用户只需要自己写一个数据接口，即可实现各种精美数据曲线的绘制，在一定程度上满足了大部分业务的需求。

本组件是MPAndroidChart的鸿蒙化版本，名为MPAndroidChart\_ohos，实现了其核心功能。

## 组件效果展示

目前MPAndroidChart\_ohos具有折线图和直方图两种图表绘制能力。下面将分别展示其折线图和直方图的绘制效果。

### 折线图

图1展示了一个由随机数据生成的折线图。MPAndroidChart\_ohos继承了原版MPAndroidChart的优秀特性，提供了多种多样的用户自定义接口，例如：

1. X、Y轴自定义。使用者可以自定义X、Y轴的位置，例如在这个sample里就绘制了左Y轴和上X轴。

2. 辅助线自定义。使用者可以选择是否显示辅助线（或格点线），也可以自由设定辅助线的位置。

3.图表美化。使用者可以设置图表曲线的各种属性（颜色、粗细等），还可以对曲线包裹区域进行填充。

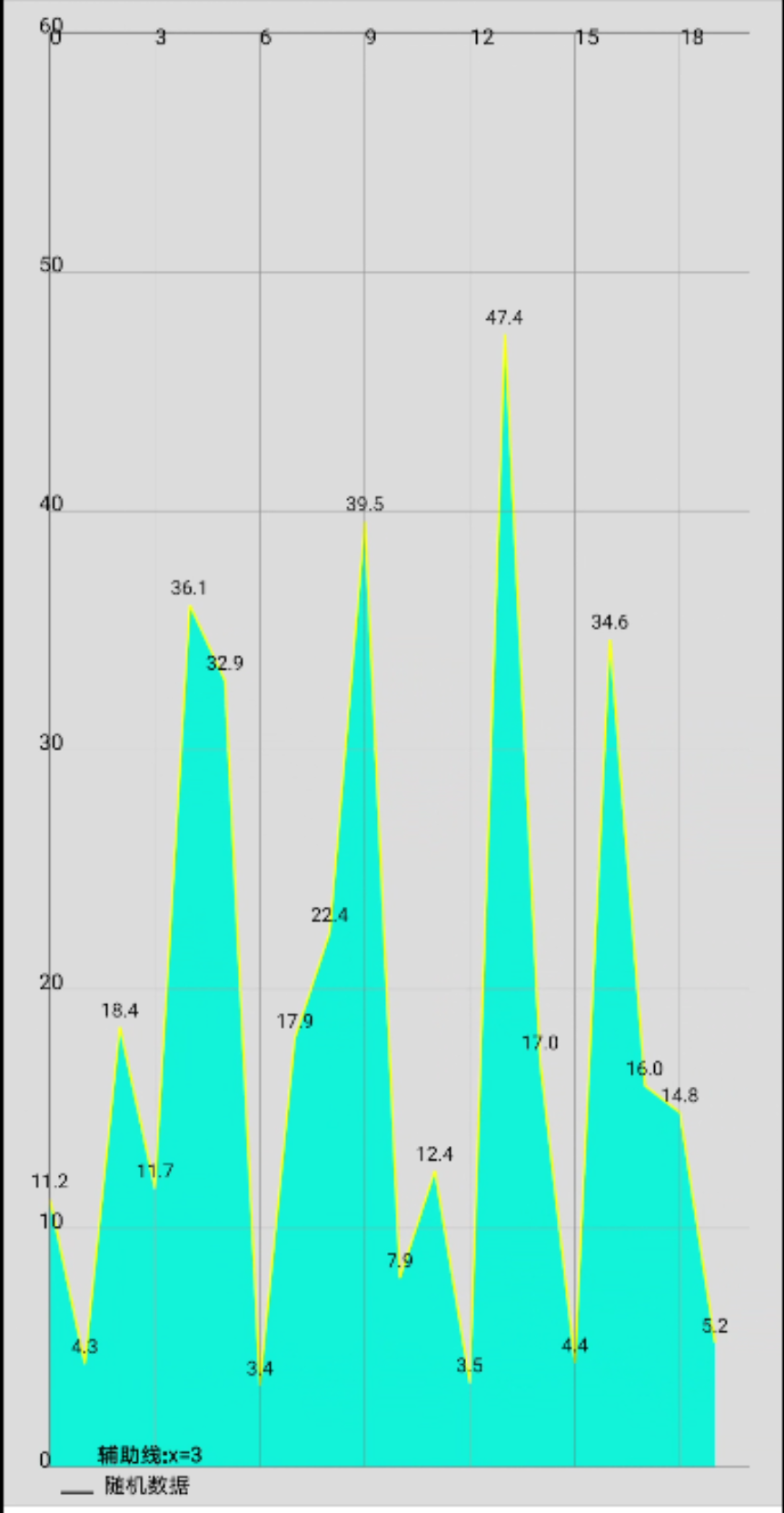


图1 折线图绘制效果

### 直方图

图2是一个假设的图表使用场景（2020年1月1日 ~ 15日的小卖部收益情况）。基于这个背景，使用MPAndroidChart\_ohos制作了一张直方图。

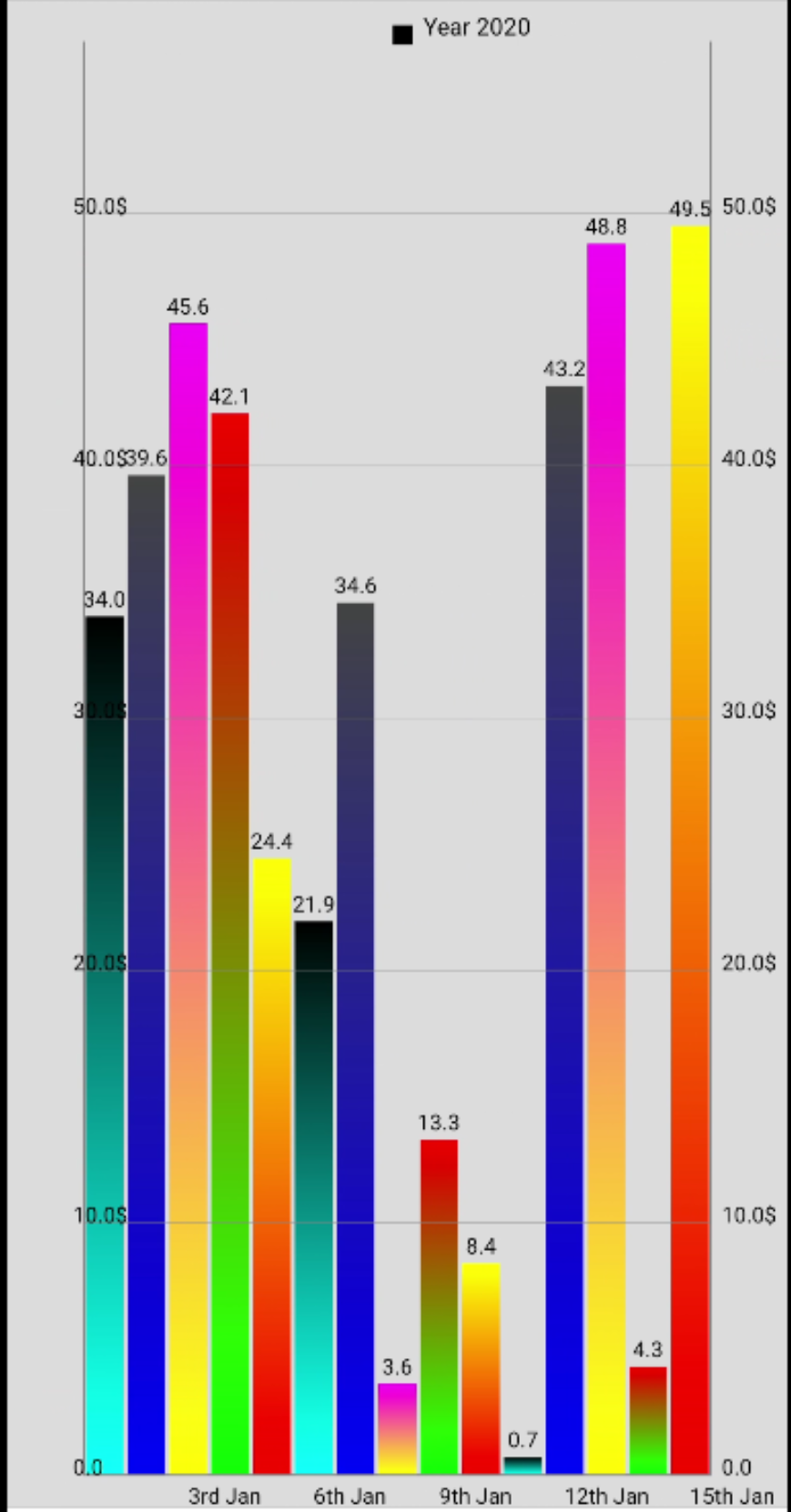


图2 直方图绘制效果

## Sample解析

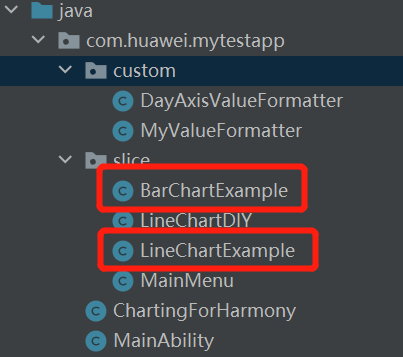


图3 Sample工程结构

图1和图2主要依靠调用Library中的能力绘制，在Sample中需要实现的部分仅在由图3红框圈出的两个文件来完成。

如果用户想要自定义绘制图表，只需要完成以下几个步骤即可：

1. 选择图表种类。
2. 设置属性。
3. 导入数据

提供了折线图和直方图的绘制能力，使用者只需要根据自身需求选择需要使用的能力即可。例如，想要绘制折线图，使用者只需要：

LineChart chart = new LineChart(context);

即可完成对折线图的初始化。同样地，直方图可以通过：

BarChart chart = new BarChart(context);

实现初始化。

### 设置属性

MPAndroidChart\_ohos提供了图表样式自定义的能力，使用者可以通过调用Library暴露的接口来给图表添加、修改、删除各项属性。例如使用者想要自定义轴线，可以通过实例化XAxis 类的对象，后通过对象的各种方法实现修改X轴的颜色，设置最大值、最小值等：

XAxis xAxis = chart.getXAxis(); // 实例化

xAxis.setAxisMaximum(20f); //属性设置

xAxis.setAxisMinimum(0f);

xAxis.setAxisLineColor(Color.BLACK.getValue());

除了轴线设置以外还可以在图表中加入各种辅助线，例如想要在x = 2处添加一条辅助线，可以通过实例化LimitLine 类的对象，后通过对象的各种方法实现修改辅助线的宽度、标签位置、文本大小等：

LimitLine llXAxis = new LimitLine(2f, "辅助线:x=2"); // 实例化

llXAxis.setLineWidth(4f); //属性设置

llXAxis.setLabelPosition(LimitLabelPosition.RIGHT\_BOTTOM);

llXAxis.setTextSize(10f);

llXAxis.setTypeface(Font.DEFAULT);

### 导入数据

在MPAndroidChart\_ohos中，不同类型的图表有着不同的数据类，例如折线图的数据类为LineData，直方图的数据类为BarData，为什么不能仅仅通过一个简单int[]或者float[]作为数据类呢？这是因为在MPAndroidChart\_ohos中数据类的作用不仅仅是承载数据，同时还需要承载一些图表相关的属性，例如曲线颜色、曲线粗细、数据点颜色、大小等，这样做的意图是在后续Library分析时会讲到。

以折线图为例，导入数据的过程如下：

（1）创建LineDataSet类：

LineDataSet set1 = new LineDataSet(values, label);

其中values是使用者想要绘制的一类数据，一般是float[]，label是这类数据的标签。

（2）将一类或者几类数据放置到一个ArrayList中：

ArrayList<ILineDataSet> dataSets = new ArrayList<>();

dataSets.add(set1);

（3）将ArrayList做成LineData数据类，并传递给chart

LineData data = new LineData(dataSets);

chart.setData(data);

## Library解析

### 工程结构对比

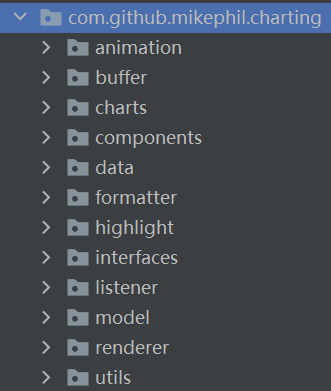
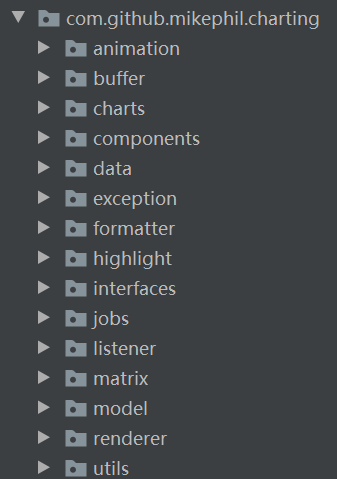
 

图 4 MPAndroidChart\_ohos(左)与MPAndroidChart (右)的工程结构对比

从图4中的两张图的对比可以看出，MPAndroidChart\_ohos是按照MPAndroidChart工程的结构开发的，实现了其主要功能，虽缺少exception、highlight、jobs这几个文件夹，并不影响主要功能的使用。

### 多设备适配

为了增加多设备适配性，MPAndroidChart内部以dp(density independent pixels)为单位来计算图表中各个部件的相对位置，在绘制图表时，统一将dp数据转化为pixel数据，这时就需要系统提供一些显示信息。在安卓中，这些信息由DisplayMetrics来提供，如下代码可以通过上下文获取到DisplayMetrics：

Resources res = context.getResources();

mMetrics = res.getDisplayMetrics();

接下来通过DisplayMetrics可以获取到屏幕的DPI，dp \* DPI即为屏幕的pixel：

public static float convertDpToPixel(float dp) {

return dp \* mMetrics.density;

}

在鸿蒙系统中，显示信息通过DisplayAttribute类来获取，以下代码可以获取到DisplayAttribute：

Display display = DisplayManager.getInstance().getDefaultDisplay(this.getContext()).get();

DisplayAttribute displayAttribute = display. getAttributes()

可以看出与安卓还是有些许不同的。得到DisplayAttribute后即可得到屏幕DPI，需要注意的是代表DPI的Java与安卓不同：

public static float convertDpToPixel(float dp) {

return dp \* mMetrics.densityPixels;

}

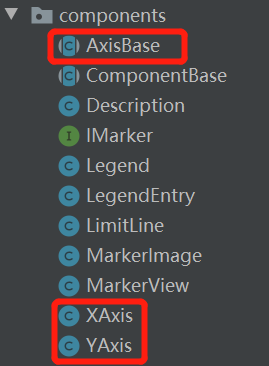
### 轴线绘制

轴线是一张图的基准，在MPAndroidChart中，轴线甚至作为了图表种类的分类基准！看似MPAndroidChart提供了十余种图表绘制的能力，其实这十余种图表是依托于两种轴线制作的，这两种轴线分别是平面直角坐标系和极坐标系。

在直角坐标系下，MPAndroidChart实现了折线图、散点图、直方图、气泡图、蜡烛图等。

在极坐标系下，MPAndroidChart实现了饼图、雷达图。

和轴线相关的类主要分布在components文件夹和renderer文件夹中：



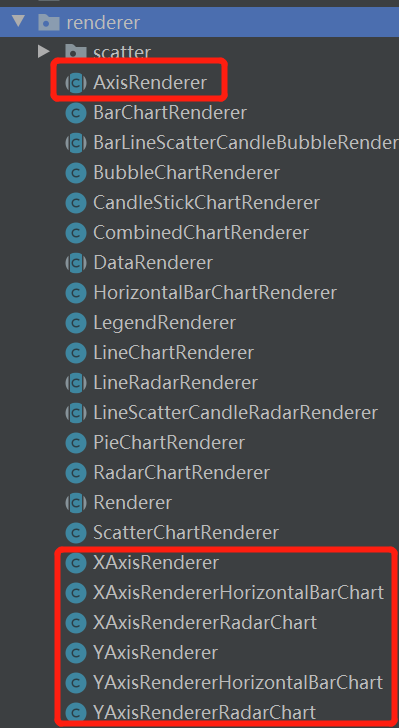


图5 轴线类与轴线绘制类

其中AxisBase类主要定义了轴应具备的属性，例如颜色、粗细、位置、刻度、标签、最值等。XAxis和YAxis继承自AxisBase，并分别定义了X、Y轴所应具备的属性，例如：X轴的位置属性应是“Top”、“BOTTOM”、“TOP\_INSIDE”、“BOTTOM\_INSIDE”或“BOTH\_SIDED”中的一种；而Y轴与X轴不同，其位置属性应为“LEFT”或“RIGHT”。

AxisRenderer类是绘制轴线的基类，其定义了绘制轴线所必备的属性和方法，例如用于绘制轴线、标签、辅助线、格点的几种画笔（Paint）和对应的方法接口。XAxisRenderer和YAxisRenderer继承自AxisRenderer，实现了其中用于绘制的接口，真正实现了轴线的绘制。其他的诸如XAxisRenderHorizontalBarChart类从名字上看也容易得知是在一些特殊图表上绘制轴线用的。

### 数据绘制

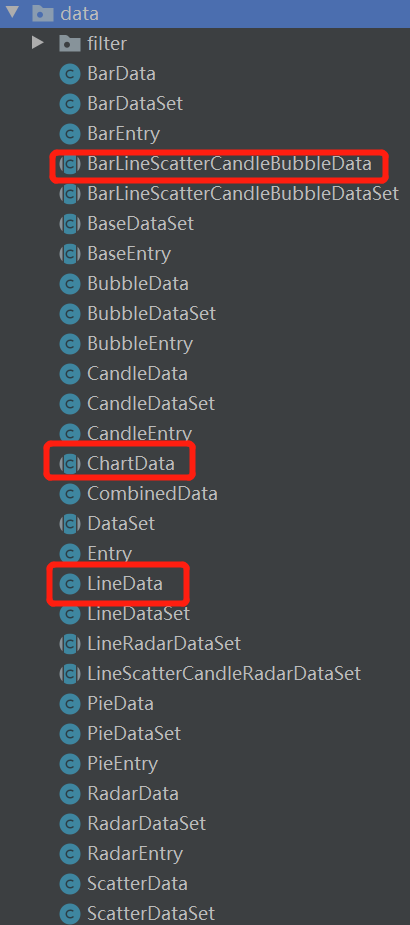


图8 折线图相关的数据类

在Sample解析中提到对于不同类型的图表，需要不同的数据类去承载数据和属性。数据类的继承关系是MPAndroidChart中比较复杂的一部分内容，举一个例子来说，我们绘制折线图所需的LineData类，它继承自：



类名有点长，不过没关系，继续向下寻找：



貌似ChartData类就是根了：



看似三级继承关系并不算多，但是值得注意的是期间需要实现的接口和泛型参数是非常多的，这些接口和泛型往往还都能继续向下嵌套好多层，这着实给移植工作带来了一些困难。下面来看看这些数据类是做什么的。

ChartData类是数据类的基类，在其中首先定义了数据的上界和下界分别是浮点数所能代表的最大和最小值，同时该类提供了一些数据处理方法，例如如果发现任何数超过了上、下界，都将这些数强制赋值为上、下界，避免溢出带来的数据错误。同时这个类还提供了诸如查询数据点个数、查询数据X、Y值、查询标签、查询最大、最小值等数据查询方法。

BarLineScatterCandleBubbleData和LineData分别是对ChartData的一次和二次封装，本身并没有添加任何方法，只是通过实现接口与各种泛型参数对存入其中的数据格式加以限制。

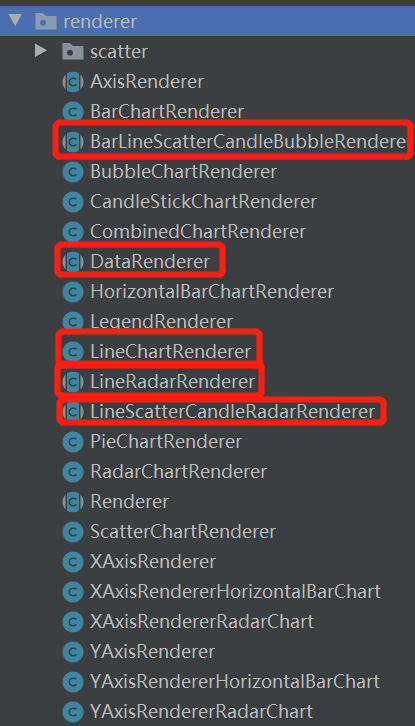


图 9 折线图的绘制类

那么数据点和曲线是如何绘制到图表中的？DataRenderer是数据绘制的基类，其中写出了绘制数据、曲线、标签等的抽象方法。继续以折线图为例，这些抽象方法将在DataRenderer🡪BarLineScatterCandleBubbleRenderer🡪LineScatterCandleRadarRenderer🡪LineRadarRenderer🡪LineChartRenderer这个继承路径中被逐步实现，最终LineChartRenderer实现了绘制折线图的全部能力。

## 项目贡献人

吴圣垚 郑森文 朱伟 陈美汝 张馨心