常用的接口：

ChartInterface

获取最大最小值，宽高，数据、中心点，最大可见值等。

BarLineScatterCandleBubbleDataProvider （继承于ChartInterface），多了getTransformer()、isInverted()、getLowestVisibleX()、getHighestVisibleX()、getData()

BarLineScatterCandleBubbleDataProvider（继承于 ChartInterface），多出来的方法有 getTransformer()、isInverted() 、getLowestVisibleX()、getHighestVisibleX()、重写了getData()

数据： ChartData（抽象方法）,里面可以有多个 IdataSet，即是说可以有多条数据，有一个重要的方法calcMinMax(),遍历获取最大最小值，这个是获得所有子数据中的最大最小值（注意与下方的IDataSet）。notifyDataSetChange()方法调用后，都会调用calcMinMax()方法。可持有多个 IdataSet的引用。

BarLineScatterCandleBubbleData LineData

IdataSet : 获取最大最小值，计算最大最小值，根据某个条件获取index，设置左边还是右边为Y轴，设置高亮，设置ValueFormatter，设置颜色，标签。

IdataSet的实现类：BaseDataSet（抽象类），dataSet标签。

转换类 Transformer

负责转换数据为像素长度

最基础的类是 Chart，继承于 ViewGroup，实现了 ChartInterface 接口

onDraw() 方法中，如果没有 数据，则在中间显示 “No data..”

如果有数据，调用 calculateOffsets() 抽象方法。

BarLineChartBase（抽象类） 继承于 Chart，实现了BarLineScatterCandleBubbleDataProvider（继承于ChartInterface），

onDraw() 方法中，

calculateOffsets()的实现是，计算偏移量。

首先执行 drawGridBackground() ，画出表格的背景，如果有边框则画边框。

然后执行 mAxisRendererLeft.computeAxis()，computeAxis()会执行 computeAxisValues() 方法<重要方法>，这个方法可以测量出 间距，还有数据的范围。

接着执行 mAxisRenderer的renderAxisLine，来画坐标轴线。

再接着执行 autoScale() ，在 mAutoScaleMinMaxEnabled 标志位是 true 的情况下。

再执行 mAxisRenderer.renderGridLines() 来画表格线。

再然后画 LimitLine，通过mAxisRenderer.renderLimitLine() 来画限制线。

clipRect()来固定画图的范围。

接着执行 mRenderer.drawData() 来画数据。 DataRenderer 是一个抽象类。

触碰事件类

ChartTouchListener（抽象类），继承于GestureDetector.SimpleOnGestureListener，里面有一个枚举，声明了图表的动作状态NONE, DRAG, X\_ZOOM, Y\_ZOOM, PINCH\_ZOOM, ROTATE, SINGLE\_TAP, DOUBLE\_TAP, LONG\_PRESS, FLING。

BarLineChartTouchListener（继承于ChartTouchListener），在这个类中实现了GestureDetector.SimpleOnGestureListener的方法。

onTouch ()方法中，获取了VelocityTracker，回调

if (mTouchMode == NONE) {

mGestureDetector.onTouchEvent(event);

}

switch (event.getAction() & MotionEvent.ACTION\_MASK) {

case MotionEvent.ACTION\_DOWN://

break;

case MotionEvent.ACTION\_POINTER\_DOWN://多点触碰——按下

break;

case MotionEvent.ACTION\_MOVE:

break;

case MotionEvent.ACTION\_UP:

break;

case MotionEvent.ACTION\_POINTER\_UP://多点触碰——抬起

break;

}

控制统计图的扩大缩小（Zoom Out缩小，Zoom In 放大），以及触碰事件，显示高亮。

包含获取多点触碰时两个点之间的距离。

触碰事件监听 OnChartGestureListener。

Void onChartGestureStart()监听动作开始

void onChartGestureEnd()监听动作结束

onChartLongPressed(MotionEvent me)长按事件

onChartDoubleTapped(MotionEvent me)双击事件

void onChartSingleTapped(MotionEvent me) 单击事件

void onChartFling(MotionEvent me1, MotionEvent me2, float velocityX, float velocityY)惯性滑动

void onChartScale(MotionEvent me, float scaleX, float scaleY)放大缩小

void onChartTranslate(MotionEvent me, float dX, float dY)拖动

可以在此设置新的 功能

IvalueFormatter 这个接口实现了 Y 轴数据的格式化。

IaxisValueFormatter 这个接口实现了 X 轴数据的格式化。

LargeValueFormatter 拥有格式化数值很大的数据（比如12345格式化为12.3k）

辅助类：

ObjectPool< T extends ObjectPool.Poolable>，用于回收 object 实例，可复用