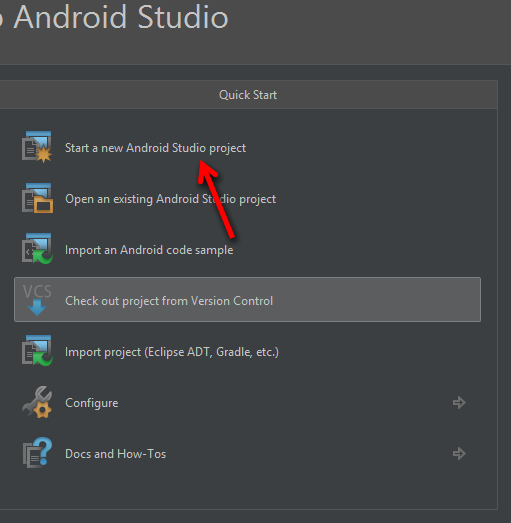
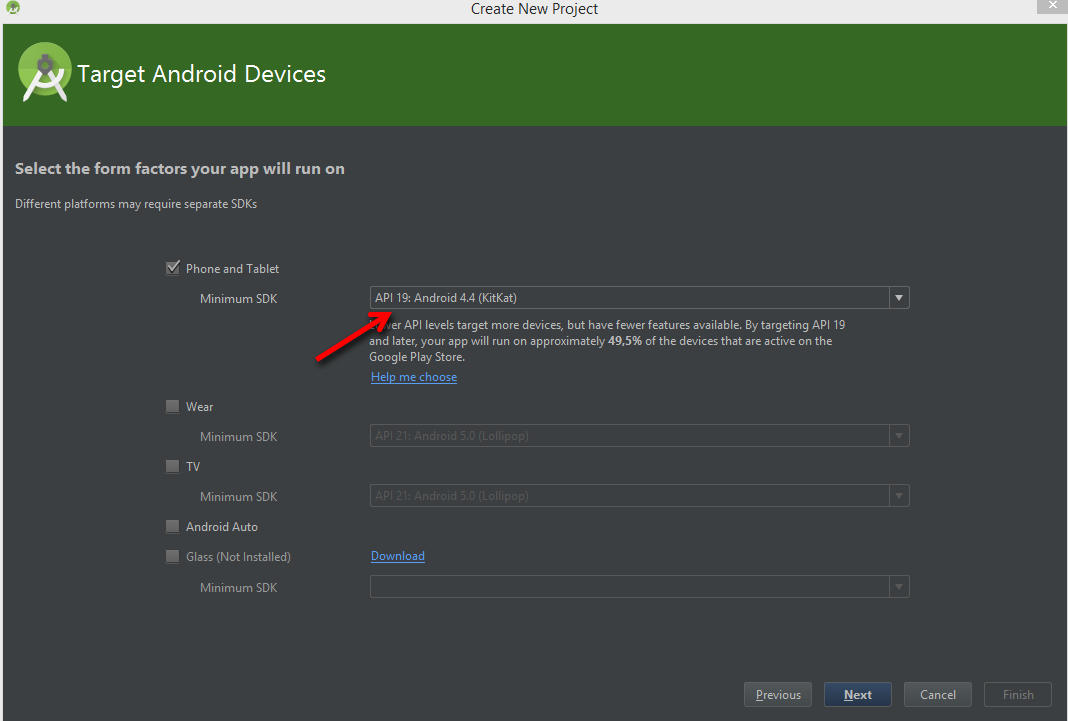
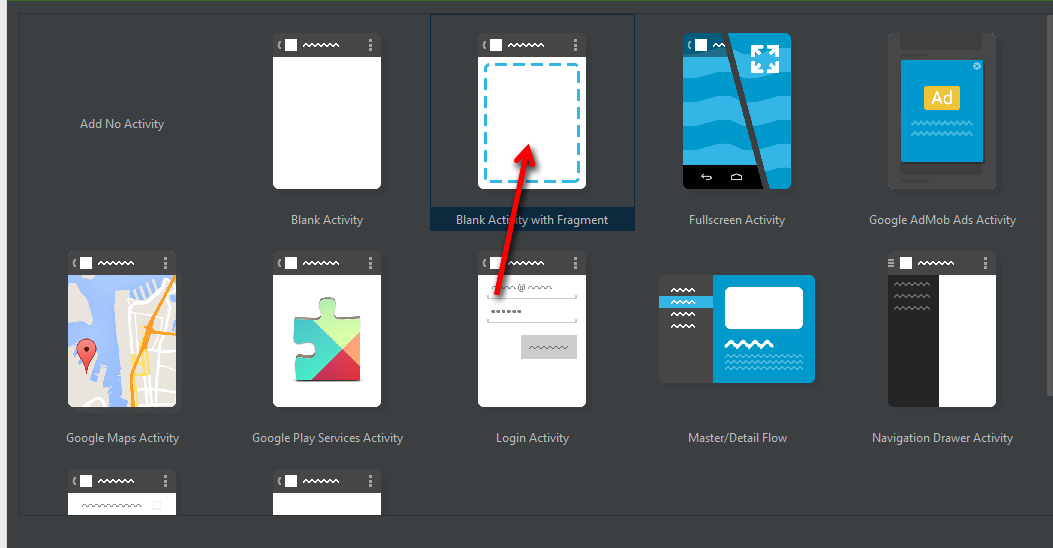
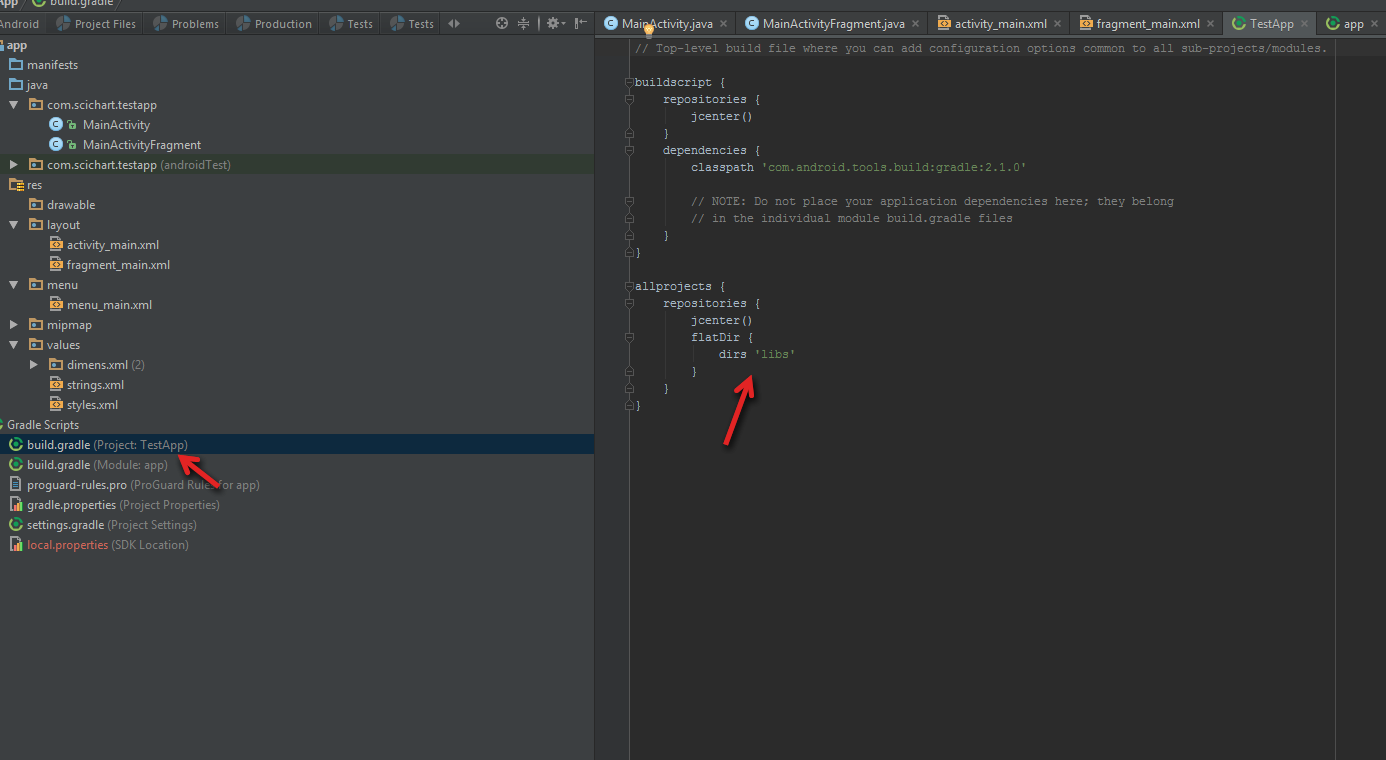
1. First of all need to create new project in Android Studio:

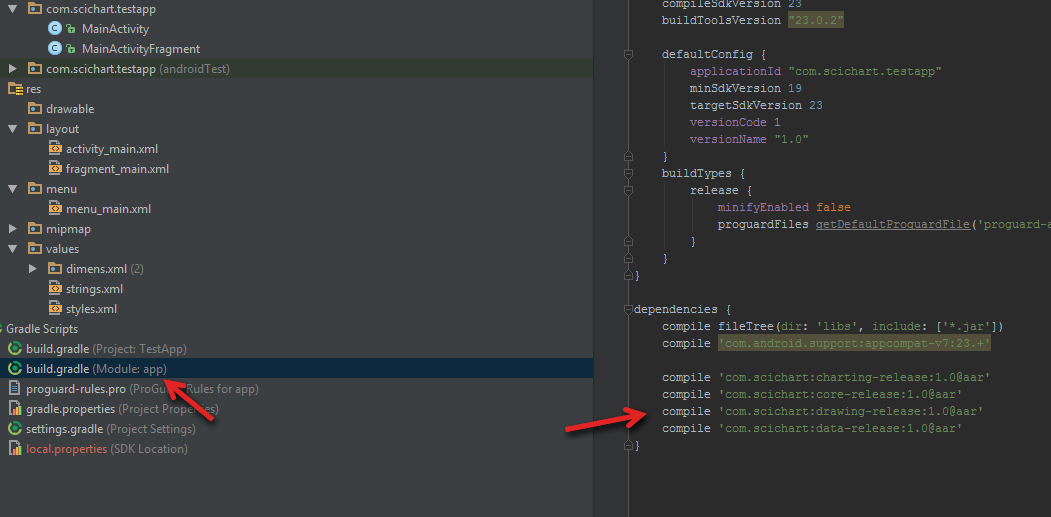


1. During creation need to select Android API 19 ( KitKat ): 
2. Choose Blanc activity with Fragment:

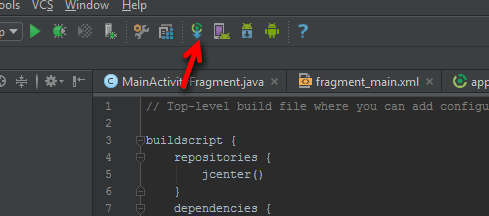


1. After creation of project need to extract zip with libraries. I extracted it into “TestApp/app/libs” folder
2. Then you need to change gradle build files:
   1. Change main build script and add “libs” folder into repositories:



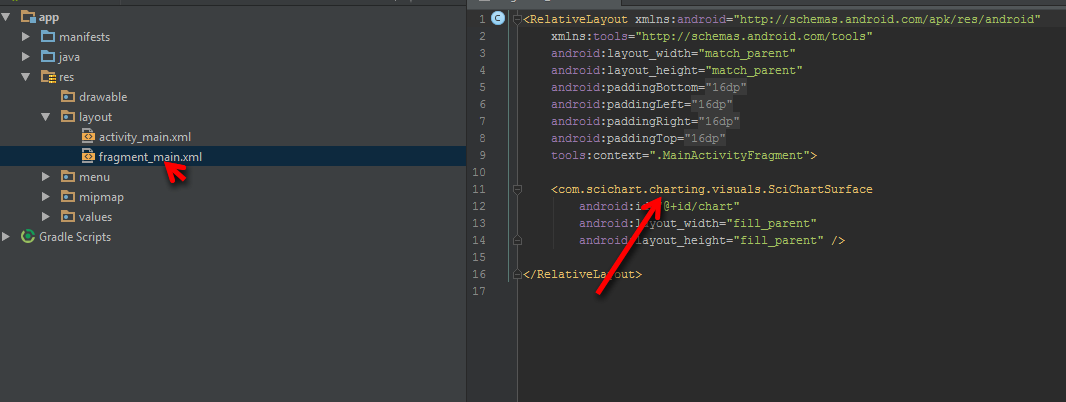
* 1. Change app build script and add aar libs as compile dependencies: 

1. Need to sync gradle config :



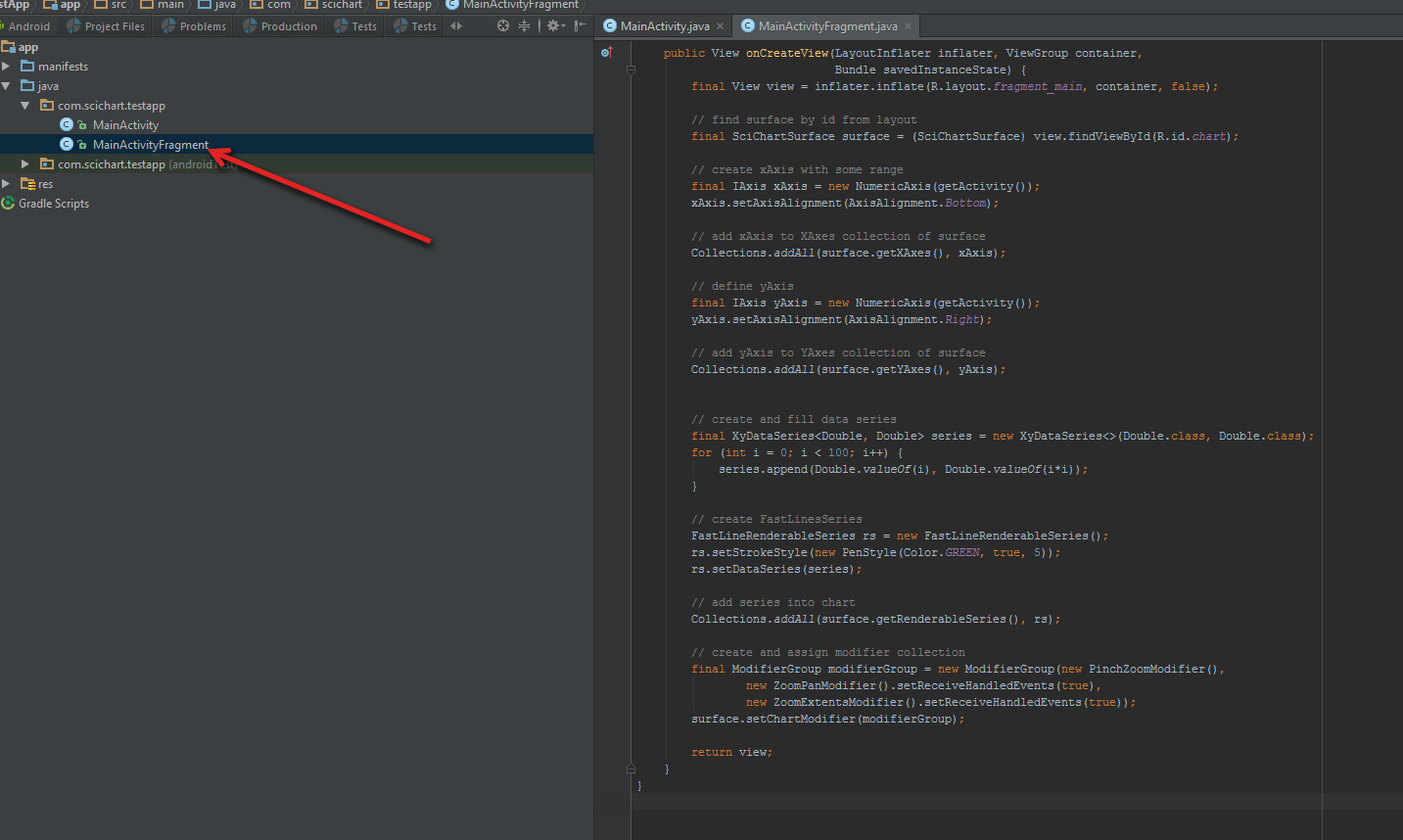
1. That’s all with configurations of project… Now you can write some code.
2. First of all need to define chart. Need to go to app/res/layout/fragment\_main.xml and add next code:

<com.scichart.charting.visuals.SciChartSurface  
 android:id="@+id/chart"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent" />



1. And then in MainActivityFragment need to define axes, renderable series or modifiers in code and set them on chart:

// find surface by id from layout  
final SciChartSurface surface = (SciChartSurface) view.findViewById(R.id.*chart*);  
  
// create xAxis with some range  
final IAxis xAxis = new NumericAxis(getActivity());  
xAxis.setAxisAlignment(AxisAlignment.*Bottom*);  
  
// add xAxis to XAxes collection of surface  
Collections.*addAll*(surface.getXAxes(), xAxis);  
  
// define yAxis  
final IAxis yAxis = new NumericAxis(getActivity());  
yAxis.setAxisAlignment(AxisAlignment.*Right*);  
  
// add yAxis to YAxes collection of surface  
Collections.*addAll*(surface.getYAxes(), yAxis);  
  
  
// create and fill data series  
final XyDataSeries<Double, Double> series = new XyDataSeries<>(Double.class, Double.class);  
for (int i = 0; i < 100; i++) {  
 series.append(Double.*valueOf*(i), Double.*valueOf*(i\*i));  
}  
  
// create FastLinesSeries  
FastLineRenderableSeries rs = new FastLineRenderableSeries();  
rs.setStrokeStyle(new PenStyle(Color.*GREEN*, true, 5));  
rs.setDataSeries(series);  
  
// add series into chart  
Collections.*addAll*(surface.getRenderableSeries(), rs);  
  
// create and assign modifier collection  
final ModifierGroup modifierGroup = new ModifierGroup(new PinchZoomModifier(),  
 new ZoomPanModifier().setReceiveHandledEvents(true),  
 new ZoomExtentsModifier().setReceiveHandledEvents(true));  
surface.setChartModifier(modifierGroup);



Then just need to run application and you should see chart on screen:

