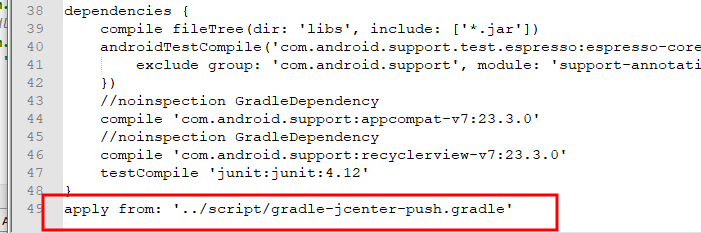
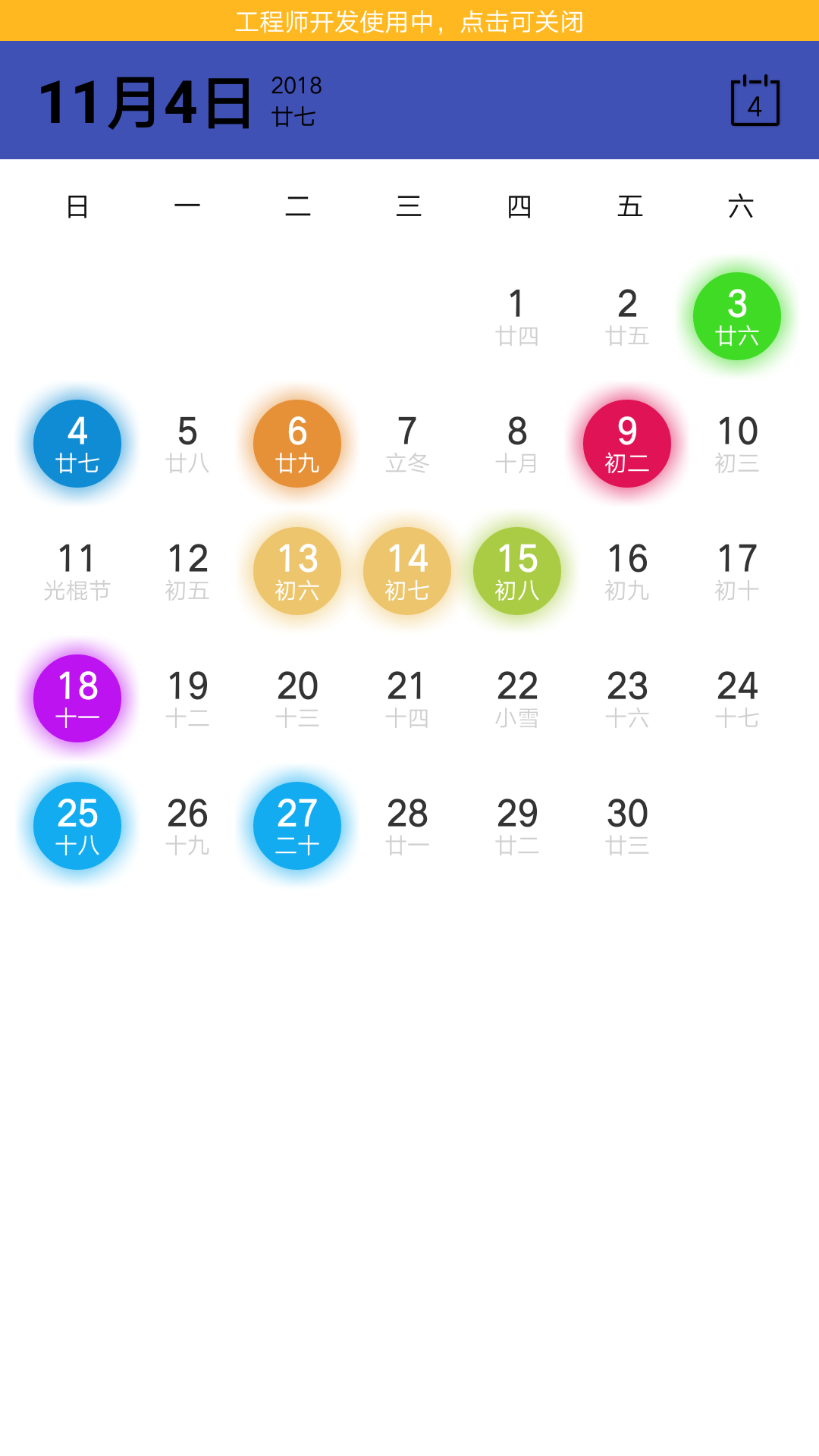
案例网址：<https://github.com/huanghaibin-dev/CalendarView>

## CalendarView日历学习

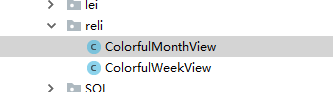
1. 导入model，导完后要把这句话删了



效果图

1. .布局

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="#ffffff"  android:fitsSystemWindows="true"  android:orientation="vertical"  tools:ignore="RtlHardcoded,HardcodedText,RtlSymmetry,SmallSp"**>  <**RelativeLayout  android:id="@+id/rl\_tool"  android:layout\_width="match\_parent"  android:layout\_height="52dp"  android:background="@color/colorPrimary"**>   <**TextView  android:id="@+id/tv\_month\_day"  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:gravity="center\_vertical"  android:paddingLeft="16dp"  android:textColor="#000000"  android:textSize="26sp"  android:textStyle="bold"  tools:ignore="RtlSymmetry"** />   <**LinearLayout  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_toRightOf="@+id/tv\_month\_day"  android:orientation="vertical"  android:paddingBottom="12dp"  android:paddingLeft="6dp"  android:paddingRight="6dp"  android:paddingTop="12dp"  tools:ignore="RelativeOverlap"**>   <**TextView  android:id="@+id/tv\_year"  android:layout\_width="wrap\_content"  android:layout\_height="0dp"  android:layout\_weight="1"  android:gravity="center"  android:textColor="#000000"  android:textSize="10sp"** />   <**TextView  android:id="@+id/tv\_lunar"  android:layout\_width="wrap\_content"  android:layout\_height="0dp"  android:layout\_weight="1"  android:gravity="center"  android:textColor="#000000"  android:textSize="10sp"  tools:ignore="SmallSp"** />  </**LinearLayout**>   <**FrameLayout  android:id="@+id/fl\_current"  android:layout\_width="32dp"  android:layout\_height="32dp"  android:layout\_alignParentRight="true"  android:layout\_centerVertical="true"  android:layout\_marginRight="12dp"  android:background="@drawable/bg\_ripple"**>   <**ImageView  android:id="@+id/ib\_calendar"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:contentDescription="@string/app\_name"  android:scaleType="centerInside"  android:src="@mipmap/ic\_calendar"  android:tint="#000000"** />   <**TextView  android:id="@+id/tv\_current\_day"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:layout\_marginTop="2dp"  android:gravity="center"  android:text="13"  android:textColor="#000000"  android:textSize="12sp"** />  </**FrameLayout**>  </**RelativeLayout**> <**com.haibin.calendarview.CalendarLayout  android:id="@+id/calendarLayout"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"  app:default\_status="shrink"  app:calendar\_show\_mode="only\_week\_view"  app:calendar\_content\_view\_id="@+id/d"** > <**com.haibin.calendarview.CalendarView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/ib\_calendarview"  android:background="#fff"  app:calendar\_padding="10dp"  app:current\_month\_lunar\_text\_color="#CFCFCF"  app:current\_month\_text\_color="#333333"  app:min\_year="2004"  app:month\_view="com.example.a11699.graduatemanager.reli.ColorfulMonthView"  app:month\_view\_show\_mode="mode\_only\_current"  app:other\_month\_text\_color="#e1e1e1"  app:scheme\_lunar\_text\_color="#ffffff"  app:scheme\_text="假"  app:scheme\_text\_color="#fff"  app:scheme\_theme\_color="#128c4b"  app:selected\_lunar\_text\_color="#fff"  app:selected\_text\_color="#fff"  app:selected\_theme\_color="#108cd4"  app:week\_background="#fff"  app:week\_text\_color="#111"  app:week\_view="com.example.a11699.graduatemanager.reli.ColorfulWeekView"  app:year\_view\_day\_text\_color="#333333"  app:year\_view\_day\_text\_size="9sp"  app:year\_view\_month\_text\_color="#ff0000"  app:year\_view\_month\_text\_size="20sp"  app:year\_view\_scheme\_color="#f17706"** > </**com.haibin.calendarview.CalendarView**> </**com.haibin.calendarview.CalendarLayout>** </**LinearLayout**> |

1. 还需要新建两个类

代碼：

ColorfulMonthView

|  |
| --- |
| **public class** ColorfulMonthView **extends** MonthView {   **private int mRadius**;   **public** ColorfulMonthView(Context context) {  **super**(context);   *//兼容硬件加速无效的代码* setLayerType(View.***LAYER\_TYPE\_SOFTWARE***, **mSelectedPaint**);  *//4.0以上硬件加速会导致无效* **mSelectedPaint**.setMaskFilter(**new** BlurMaskFilter(30, BlurMaskFilter.Blur.***SOLID***));   setLayerType(View.***LAYER\_TYPE\_SOFTWARE***, **mSchemePaint**);  **mSchemePaint**.setMaskFilter(**new** BlurMaskFilter(30, BlurMaskFilter.Blur.***SOLID***));  }   @Override  **protected void** onPreviewHook() {  **mRadius** = Math.*min*(**mItemWidth**, **mItemHeight**) / 5 \* 2;  }   */\*\*  \* 如果需要点击Scheme没有效果，则return true  \*  \** ***@param canvas*** *canvas  \** ***@param calendar*** *日历日历calendar  \** ***@param x*** *日历Card x起点坐标  \** ***@param y*** *日历Card y起点坐标  \** ***@param hasScheme*** *hasScheme 非标记的日期  \** ***@return*** *false 则不绘制onDrawScheme，因为这里背景色是互斥的  \*/* @Override  **protected boolean** onDrawSelected(Canvas canvas, Calendar calendar, **int** x, **int** y, **boolean** hasScheme) {  **int** cx = x + **mItemWidth** / 2;  **int** cy = y + **mItemHeight** / 2;  canvas.drawCircle(cx, cy, **mRadius**, **mSelectedPaint**);  **return true**;  }   @Override  **protected void** onDrawScheme(Canvas canvas, Calendar calendar, **int** x, **int** y) {  **int** cx = x + **mItemWidth** / 2;  **int** cy = y + **mItemHeight** / 2;  canvas.drawCircle(cx, cy, **mRadius**, **mSchemePaint**);  }   @Override  **protected void** onDrawText(Canvas canvas, Calendar calendar, **int** x, **int** y, **boolean** hasScheme, **boolean** isSelected) {  **int** cx = x + **mItemWidth** / 2;  **int** top = y - **mItemHeight** / 8;  **if** (isSelected) {  canvas.drawText(String.*valueOf*(calendar.getDay()), cx, **mTextBaseLine** + top,  **mSelectTextPaint**);  canvas.drawText(calendar.getLunar(), cx, **mTextBaseLine** + y + **mItemHeight** / 10, **mSelectedLunarTextPaint**);  } **else if** (hasScheme) {  canvas.drawText(String.*valueOf*(calendar.getDay()), cx, **mTextBaseLine** + top,  calendar.isCurrentDay() ? **mCurDayTextPaint** :  calendar.isCurrentMonth() ? **mSchemeTextPaint** : **mOtherMonthTextPaint**);   canvas.drawText(calendar.getLunar(), cx, **mTextBaseLine** + y + **mItemHeight** / 10, **mSchemeLunarTextPaint**);  } **else** {  canvas.drawText(String.*valueOf*(calendar.getDay()), cx, **mTextBaseLine** + top,  calendar.isCurrentDay() ? **mCurDayTextPaint** :  calendar.isCurrentMonth() ? **mCurMonthTextPaint** : **mOtherMonthTextPaint**);  canvas.drawText(calendar.getLunar(), cx, **mTextBaseLine** + y + **mItemHeight** / 10, **mCurMonthLunarTextPaint**);  }  } } |

ColorfulWeekView类

|  |
| --- |
| **public class** ColorfulWeekView **extends** WeekView {   **private int mRadius**;   **public** ColorfulWeekView(Context context) {  **super**(context);  *//兼容硬件加速无效的代码* setLayerType(View.***LAYER\_TYPE\_SOFTWARE***, **mSelectedPaint**);  *//4.0以上硬件加速会导致无效* **mSelectedPaint**.setMaskFilter(**new** BlurMaskFilter(30, BlurMaskFilter.Blur.***SOLID***));   setLayerType(View.***LAYER\_TYPE\_SOFTWARE***, **mSchemePaint**);  **mSchemePaint**.setMaskFilter(**new** BlurMaskFilter(30, BlurMaskFilter.Blur.***SOLID***));  }   @Override  **protected void** onPreviewHook() {  **mRadius** = Math.*min*(**mItemWidth**, **mItemHeight**) / 5 \* 2;  }   */\*\*  \* 如果需要点击Scheme没有效果，则return true  \*  \** ***@param canvas*** *canvas  \** ***@param calendar*** *日历日历calendar  \** ***@param x*** *日历Card x起点坐标  \** ***@param hasScheme*** *hasScheme 非标记的日期  \** ***@return*** *false 则不绘制onDrawScheme，因为这里背景色是互斥的  \*/* @Override  **protected boolean** onDrawSelected(Canvas canvas, Calendar calendar, **int** x, **boolean** hasScheme) {  **int** cx = x + **mItemWidth** / 2;  **int** cy = **mItemHeight** / 2;  canvas.drawCircle(cx, cy, **mRadius**, **mSelectedPaint**);  **return true**;  }    @Override  **protected void** onDrawScheme(Canvas canvas, Calendar calendar, **int** x) {  **int** cx = x + **mItemWidth** / 2;  **int** cy = **mItemHeight** / 2;  canvas.drawCircle(cx, cy, **mRadius**, **mSchemePaint**);  }   @Override  **protected void** onDrawText(Canvas canvas, Calendar calendar, **int** x, **boolean** hasScheme, **boolean** isSelected) {  **int** cx = x + **mItemWidth** / 2;  **int** top = -**mItemHeight** / 8;  **if** (isSelected) {  canvas.drawText(String.*valueOf*(calendar.getDay()), cx, **mTextBaseLine** + top,  **mSelectTextPaint**);  canvas.drawText(calendar.getLunar(), cx, **mTextBaseLine** + **mItemHeight** / 10, **mSelectedLunarTextPaint**);  } **else if** (hasScheme) {  canvas.drawText(String.*valueOf*(calendar.getDay()), cx, **mTextBaseLine** + top,  calendar.isCurrentDay() ? **mCurDayTextPaint** :  calendar.isCurrentMonth() ? **mSchemeTextPaint** : **mSchemeTextPaint**);   canvas.drawText(calendar.getLunar(), cx, **mTextBaseLine** + **mItemHeight** / 10, **mSchemeLunarTextPaint**);  } **else** {  canvas.drawText(String.*valueOf*(calendar.getDay()), cx, **mTextBaseLine** + top,  calendar.isCurrentDay() ? **mCurDayTextPaint** :  calendar.isCurrentMonth() ? **mCurMonthTextPaint** : **mCurMonthTextPaint**);  canvas.drawText(calendar.getLunar(), cx, **mTextBaseLine** + **mItemHeight** / 10, **mCurMonthLunarTextPaint**);  }  } } |

在新建一个基类jilei.java

|  |
| --- |
| **package** com.example.a11699.graduatemanager.activity;  **import** android.annotation.SuppressLint; **import** android.os.Build; **import** android.os.Bundle; **import** android.support.annotation.Nullable; **import** android.support.v7.app.AppCompatActivity; **import** android.view.View; **import** android.view.Window; **import** android.view.WindowManager;  **import** java.lang.reflect.Field; **import** java.lang.reflect.Method; **public abstract class** jilei **extends** AppCompatActivity {   **private static boolean** *isMiUi* = **false**;   **protected void** initWindow() {   }   **protected abstract int** getLayoutId();   **protected abstract void** initView();   **protected abstract void** initData();   @Override  **protected void** onCreate(@Nullable Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  initWindow();  setContentView(getLayoutId());  initView();  initData();  }   */\*\*  \* 设置小米黑色状态栏字体  \*/* @SuppressLint(**"PrivateApi"**)  **void** setMIUIStatusBarDarkMode() {  **if** (*isMiUi*) {  Class<? **extends** Window> clazz = getWindow().getClass();  **try** {  **int** darkModeFlag;  Class<?> layoutParams = Class.*forName*(**"android.view.MiuiWindowManager$LayoutParams"**);  Field field = layoutParams.getField(**"EXTRA\_FLAG\_STATUS\_BAR\_DARK\_MODE"**);  darkModeFlag = field.getInt(layoutParams);  Method extraFlagField = clazz.getMethod(**"setExtraFlags"**, **int**.**class**, **int**.**class**);  extraFlagField.invoke(getWindow(), darkModeFlag, darkModeFlag);  } **catch** (Exception e) {  e.printStackTrace();  }  }  }   */\*  \* 静态域，获取系统版本是否基于MIUI  \*/* **static** {  **try** {  @SuppressLint(**"PrivateApi"**) Class<?> sysClass = Class.*forName*(**"android.os.SystemProperties"**);  Method getStringMethod = sysClass.getDeclaredMethod(**"get"**, String.**class**);  String version = (String) getStringMethod.invoke(sysClass, **"ro.miui.ui.version.name"**);  *isMiUi* = version.compareTo(**"V6"**) >= 0 && Build.VERSION.***SDK\_INT*** < 24;  } **catch** (Exception e) {  e.printStackTrace();  }  }    */\*\*  \* 设置魅族手机状态栏图标颜色风格  \* 可以用来判断是否为Flyme用户  \*  \** ***@param window*** *需要设置的窗口  \** ***@param dark*** *是否把状态栏字体及图标颜色设置为深色  \** ***@return*** *boolean 成功执行返回true  \*/* **public static boolean** setMeiZuDarkMode(Window window, **boolean** dark) {  **boolean** result = **false**;  **if** (Build.VERSION.***SDK\_INT*** >= 24) {  **return false**;  }  **if** (window != **null**) {  **try** {  WindowManager.LayoutParams lp = window.getAttributes();  Field darkFlag = WindowManager.LayoutParams.**class** .getDeclaredField(**"MEIZU\_FLAG\_DARK\_STATUS\_BAR\_ICON"**);  Field meizuFlags = WindowManager.LayoutParams.**class** .getDeclaredField(**"meizuFlags"**);  darkFlag.setAccessible(**true**);  meizuFlags.setAccessible(**true**);  **int** bit = darkFlag.getInt(**null**);  **int** value = meizuFlags.getInt(lp);  **if** (dark) {  value |= bit;  } **else** {  value &= ~bit;  }  meizuFlags.setInt(lp, value);  window.setAttributes(lp);  result = **true**;  } **catch** (Exception e) {  e.printStackTrace();  }  }  **return** result;  }   @SuppressLint(**"InlinedApi"**)  **private int** getStatusBarLightMode() {  **int** result = 0;  **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***KITKAT***) {  **if** (*isMiUi*) {  result = 1;  } **else if** (*setMeiZuDarkMode*(getWindow(), **true**)) {  result = 2;  } **else if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***M***) {  getWindow().getDecorView().setSystemUiVisibility(View.***SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN*** | View.***SYSTEM\_UI\_FLAG\_LIGHT\_STATUS\_BAR***);  result = 3;  }  }  **return** result;  }    @SuppressLint(**"InlinedApi"**)  **protected void** setStatusBarDarkMode() {  **int** type = getStatusBarLightMode();  **if** (type == 1) {  setMIUIStatusBarDarkMode();  } **else if** (type == 2) {  *setMeiZuDarkMode*(getWindow(), **true**);  } **else if** (type == 3) {  getWindow().getDecorView().setSystemUiVisibility(View.***SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN*** | View.***SYSTEM\_UI\_FLAG\_LIGHT\_STATUS\_BAR***);  }  } } |

绘制日历

|  |
| --- |
| **public class** writezhouji **extends** jilei **implements** CalendarView.OnCalendarSelectListener, CalendarView.OnYearChangeListener, View.OnClickListener {    TextView **mTextMonthDay**;   TextView **mTextYear**;   TextView **mTextLunar**;   TextView **mTextCurrentDay**;    CalendarView **mCalendarView**;  RelativeLayout **mRelativeTool**;  **private int mYear**;  CalendarLayout **mCalendarLayout**;   @Override  **protected int** getLayoutId() {  **return** R.layout.***activity\_writezhouji***;  }   @Override  **protected void** initView() {  initview();   }   @Override  **protected void** initData() {  **int** year = **mCalendarView**.getCurYear();  **int** month = **mCalendarView**.getCurMonth();   Map<String, Calendar> map = **new** HashMap<>();  map.put(getSchemeCalendar(year, month, 3, 0xFF40db25, **"假"**).toString(),  getSchemeCalendar(year, month, 3, 0xFF40db25, **"假"**));  map.put(getSchemeCalendar(year, month, 6, 0xFFe69138, **"事"**).toString(),  getSchemeCalendar(year, month, 6, 0xFFe69138, **"事"**));  map.put(getSchemeCalendar(year, month, 9, 0xFFdf1356, **"议"**).toString(),  getSchemeCalendar(year, month, 9, 0xFFdf1356, **"议"**));  map.put(getSchemeCalendar(year, month, 13, 0xFFedc56d, **"记"**).toString(),  getSchemeCalendar(year, month, 13, 0xFFedc56d, **"记"**));  map.put(getSchemeCalendar(year, month, 14, 0xFFedc56d, **"记"**).toString(),  getSchemeCalendar(year, month, 14, 0xFFedc56d, **"记"**));  map.put(getSchemeCalendar(year, month, 15, 0xFFaacc44, **"假"**).toString(),  getSchemeCalendar(year, month, 15, 0xFFaacc44, **"假"**));  map.put(getSchemeCalendar(year, month, 18, 0xFFbc13f0, **"记"**).toString(),  getSchemeCalendar(year, month, 18, 0xFFbc13f0, **"记"**));  map.put(getSchemeCalendar(year, month, 25, 0xFF13acf0, **"假"**).toString(),  getSchemeCalendar(year, month, 25, 0xFF13acf0, **"假"**));  map.put(getSchemeCalendar(year, month, 27, 0xFF13acf0, **"多"**).toString(),  getSchemeCalendar(year, month, 27, 0xFF13acf0, **"多"**));  *//此方法在巨大的数据量上不影响遍历性能，推荐使用* **mCalendarView**.setSchemeDate(map);  }  **private** Calendar getSchemeCalendar(**int** year, **int** month, **int** day, **int** color, String text) {  Calendar calendar = **new** Calendar();  calendar.setYear(year);  calendar.setMonth(month);  calendar.setDay(day);  calendar.setSchemeColor(color);*//如果单独标记颜色、则会使用这个颜色* calendar.setScheme(text);  **return** calendar;  }   **void** initview() {  setStatusBarDarkMode();  **mCalendarView**=findViewById(R.id.***ib\_calendarview***);  **mTextMonthDay** = (TextView) findViewById(R.id.***tv\_month\_day***);  **mTextYear** = (TextView) findViewById(R.id.***tv\_year***);  **mTextLunar** = (TextView) findViewById(R.id.***tv\_lunar***);  **mRelativeTool** = (RelativeLayout) findViewById(R.id.***rl\_tool***);  **mTextCurrentDay** = (TextView) findViewById(R.id.***tv\_current\_day***);  *//这个是左上角的控件* **mTextMonthDay**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **if** (!**mCalendarLayout**.isExpand()) {  **mCalendarView**.showYearSelectLayout(**mYear**);*////快速弹出年份选择月份* **return**;  }  **mCalendarView**.showYearSelectLayout(**mYear**);  **mTextLunar**.setVisibility(View.***GONE***);  **mTextYear**.setVisibility(View.***GONE***);  **mTextMonthDay**.setText(String.*valueOf*(**mYear**));  }  });  *//这个是右上角的控件* findViewById(R.id.***fl\_current***).setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **mCalendarView**.scrollToCurrent();*////滚动到当前日期* }  });  **mCalendarLayout** = (CalendarLayout) findViewById(R.id.***calendarLayout***);  **mCalendarView**.setOnCalendarSelectListener(**this**);  **mCalendarView**.setOnYearChangeListener(**this**);  **mTextYear**.setText(String.*valueOf*(**mCalendarView**.getCurYear()));  **mYear** = **mCalendarView**.getCurYear();  **mTextMonthDay**.setText(**mCalendarView**.getCurMonth() + **"月"** + **mCalendarView**.getCurDay() + **"日"**);  **mTextLunar**.setText(**"今日"**);  **mTextCurrentDay**.setText(String.*valueOf*(**mCalendarView**.getCurDay()));  }   @Override  **public void** onClick(View v) {   }   @Override  **public void** onCalendarOutOfRange(Calendar calendar) {   } *//日历被点击的时候* @Override  **public void** onCalendarSelect(Calendar calendar, **boolean** isClick) {  **mTextLunar**.setVisibility(View.***VISIBLE***);  **mTextYear**.setVisibility(View.***VISIBLE***);  **mTextMonthDay**.setText(calendar.getMonth() + **"月"** + calendar.getDay() + **"日"**);  **mTextYear**.setText(String.*valueOf*(calendar.getYear()));  **mTextLunar**.setText(calendar.getLunar());  **mYear** = calendar.getYear();  }   @Override  **public void** onYearChange(**int** year) {  **mTextMonthDay**.setText(String.*valueOf*(year));  } |