



# Junior Android Developer

24 октября 2013 г.

## 1 Drawables

## 2 Adapter

## 3 ViewPager и PagerAdapter

- Это абстракция того, что может быть показана на экране (нарисовано)
- Находиться в пакете **android.graphics.drawable**:  
PictureDrawable, GradientDrawable, LayerDrawable ...
- Ресурсы в папке **res/drawable**: PNG, JPG, GIF и XML

## Из кода

```
ImageView image =  
    new ImageView(this);  
image  
    .setImageResource(R.drawable.image);  
...  
Resources res =  
    context.getResources();  
Drawable image2 =  
    res.getDrawable(R.drawable.image2);
```

## Из XML

```
<ImageView  
    ...  
    android:src=  
        "@drawable/image"  
/>
```

- Для каждого файла помещенного в папку res/drawable создается автоматически
- Можно создать из xml

```
<?xml version="1.0" encoding="utf-8"?>
<bitmap
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:src="@[package:]drawable/drawable_resource"
    android:antialias=["true" | "false"]
    android:dither=["true" | "false"]
    android:filter=["true" | "false"]
    android:gravity=["top" | "bottom" | "left" | "right" | "center_vertical" |
        "fill_vertical" | "center_horizontal" | "fill_horizontal" |
        "center" | "fill" | "clip_vertical" | "clip_horizontal"]
    android:mipMap=["true" | "false"]
    android:tileMode=["disabled" | "clamp" | "repeat" | "mirror"] />
```

- Описание геометрической фигуры
- Задаем форму, контур, цвет и градиент

## Результат



## XML

```
<?xml version="1.0" encoding="utf-8"?>
<shape
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:shape="rectangle">
  <gradient
    android:startColor="#FFFF0000"
    android:endColor="#80FF00FF"
    android:angle="45"/>
  <padding android:left="7dp"
    android:top="7dp"
    android:right="7dp"
    android:bottom="7dp" />
  <corners android:radius="8dp" />
</shape>
```

- Используя состояния автоматически подставляет нужную Drawable.

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android"
    android:constantSize=["true" | "false"]
    android:dither=["true" | "false"]
    android:variablePadding=["true" | "false"] >
    <item
        android:drawable="@[package:]drawable/drawable_resource"
        android:state_pressed=["true" | "false"]
        android:state_focused=["true" | "false"]
        android:state_hovered=["true" | "false"]
        android:state_selected=["true" | "false"]
        android:state_checkable=["true" | "false"]
        android:state_checked=["true" | "false"]
        android:state_enabled=["true" | "false"]
        android:state_activated=["true" | "false"]
        android:state_window_focused=["true" | "false"] />
</selector>
```

- Не растр а “тянучка”
- Инструмент **draw9path**
- Верх отвечает за масштабируемость по горизонтали
- Левая сторона отвечает за масштабируемость по вертикали
- Правая и нижняя стороны отвечают за расположение содержимого

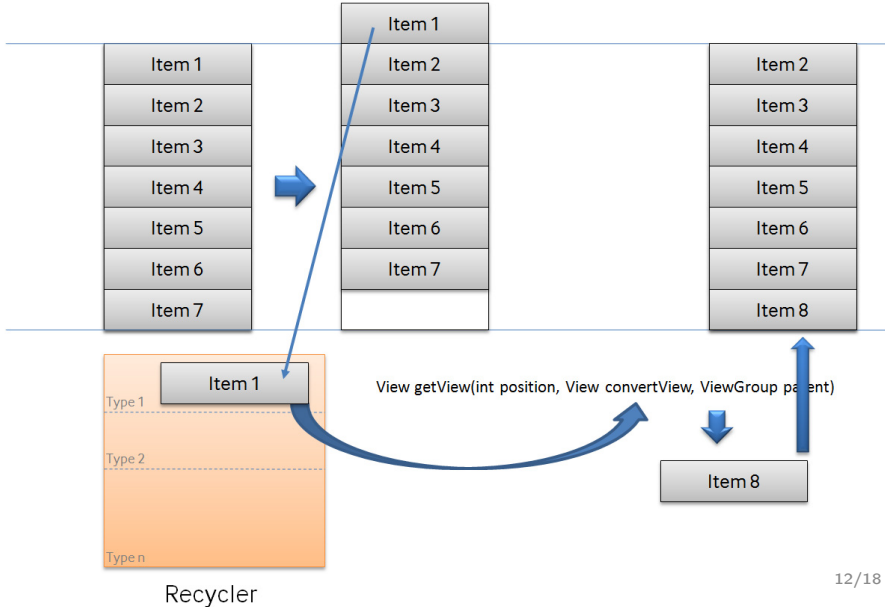


- Нет возможности описать динамическое содержимое с помощью обычных Layout
- **пример:** Вывод списка из БД
- Из-за ограничений мы не можем выделять память сразу под большой объем данных
- **пример:** Записей всего 10000 а на экран помещается всего 10

- Специальное View, содержимое которого определяется динамически с помощью специального объекта-адаптера
- ListView, ExpandableListView, GridView ...
- Мы должны зарегистрировать Adapter <http://developer.android.com/guide/topics/ui/binding.html>

- Является связующим элементом между AdapterView и данными
- Создает View для каждого элемента из данных (обычно это набор)
- ArrayAdapter, BaseAdapter, SimpleCursorAdapter, ...

# Повторное использование



```
public View getView(int position, View convertView,
    ViewGroup parent) {
    if (convertView == null) {
        convertView =
            inflater.inflate(R.layout.layout, null);
    }

    TextView text =
        (TextView) convertView.findViewById(R.id.text);
    text.setText("Position_" + position);

    return convertView;
}
```

```
public View getView(int position, View convertView, ViewGroup parent) {
    ViewHolder holder;

    if (convertView == null) {
        convertView = inflater.inflate(R.layout.layout, null);

        holder = new ViewHolder();
        holder.text =
            (TextView) convertView.findViewById(R.id.text);

        convertView.setTag(holder);
    } else {
        holder = (ViewHolder) convertView.getTag();
    }

    holder.text.setText("Position_" + position);
    return convertView;
}

private static class ViewHolder {
    public TextView text;
}
```

- Создаем ListView

```
ListView listView = new ListView();
```

- Создаем адаптер и регистрируем его

```
ContactAdapter a = new ContactAdapter();  
listView.setAdapter(a);
```

- Все манипуляции с данными делаем через адаптер

```
a.notifyDataSetChanged();  
a.notifyDataSetInvalidated();  
a.clear();  
a.getCount();
```

```
public class ArticleAdapter extends PagerAdapter {

    @Override
    public int getCount() {
        return dataLength < 2 ? dataLength : 1000000;
    }

    @Override
    public boolean isViewFromObject(View view, Object object) {
        return view == object;
    }

    @Override
    public Object instantiateItem(ViewGroup container, int position) {
        view = inflater.inflate(R.layout.layout, null);
        container.addView(view);
        return view;
    }

    @Override
    public void destroyItem(ViewGroup container, int position, Object object) {
        container.removeView((View)object);
    }
}
```



- Указываем в xml и не забываем указывать полный путь

```
<android.support.v4.view.ViewPager  
    android:id="@+id/pager"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
</android.support.v4.view.ViewPager>
```

- Создаем адаптер и регистрируем его

```
ArticleAdapter a = new ArticleAdapter();  
findViewById(R.id.pager).setAdapter(a);
```

- Все манипуляции с данными делаются самостоятельно через адаптер

# В следующей лекции

---

- AsyncTask
- TaskLoader
- Разберем проблемы синхронизации между компонентами