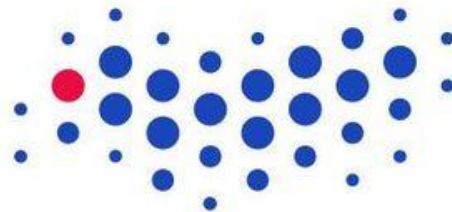


# Практикум на ЭВМ.

Программирование под Android

## Занятие 4



# LayoutInflater

# LayoutInflater

- Создаёт иерархию объектов интерфейса из файла ресурса.

```
LayoutInflater inflater = (LayoutInflater)context.  
    getSystemService(Context.LAYOUT_INFLATER_SERVICE);  
  
View view = inflater.inflate(R.layout.item_list, null, false);
```

# Списки

# ScrollView

- Данные не помещаются на экран
- Только 1 дочерний элемент

# ScrollView

```
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </LinearLayout>
</ScrollView>
```

# ScrollView

```
group = (ViewGroup)findViewById(R.id.root_container);

for (int i = 0; i < 10_000; i++) {
    TextView tv = new TextView(this);
    tv.setText("Item #" + i);

    group.addView(tv, new ViewGroup.LayoutParams(
        ViewGroup.LayoutParams.MATCH_PARENT, 80));
}
```

# ScrollView

- Много данных:
  - Медленно
  - Много ОЗУ

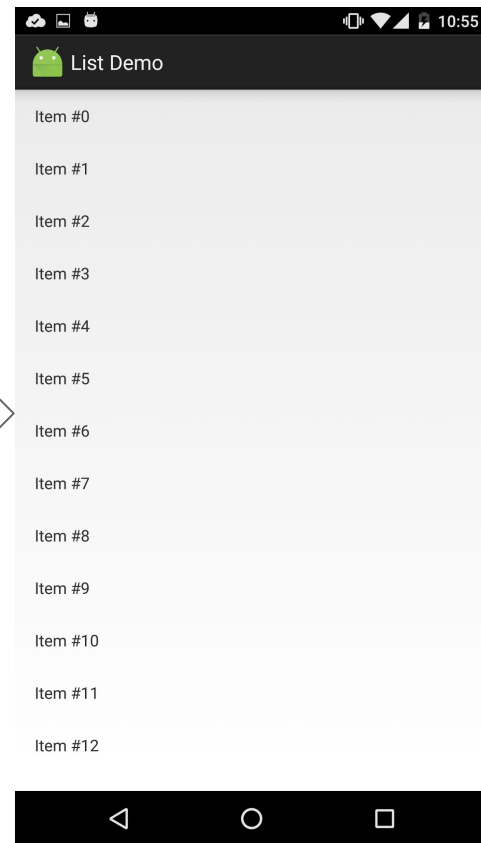
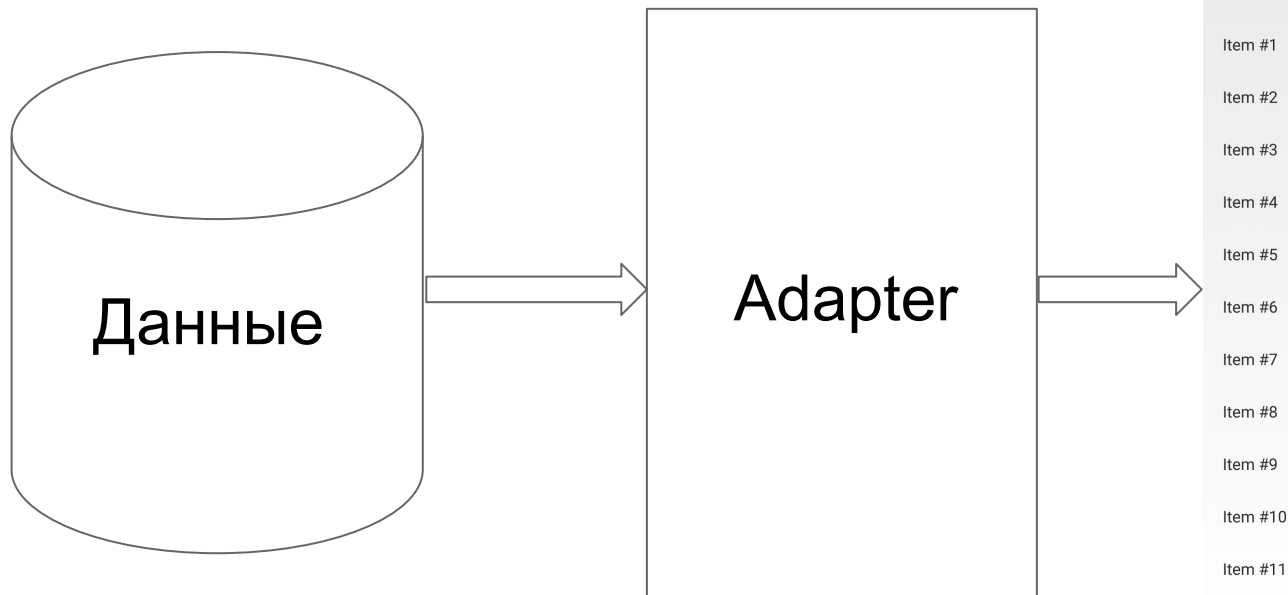


# RecyclerView

- Пришёл на смену ListView
- build.gradle:

```
dependencies {  
    compile "com.android.support:recyclerview-v7:23.1.0"  
}
```

# Adapter



# Adapter

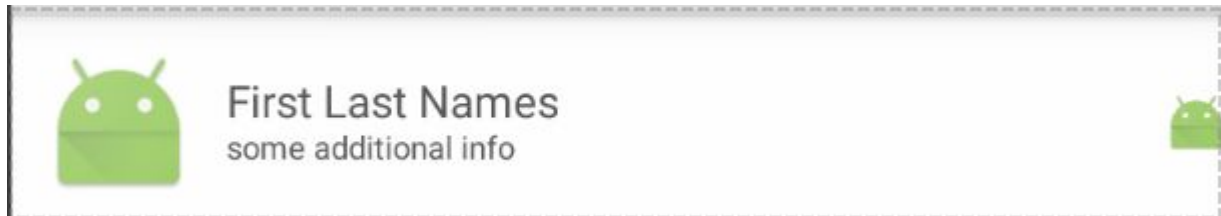
```
public static abstract class Adapter<VH extends ViewHolder>
```

```
private class SimpleRecyclerAdapter extends
```

```
RecyclerView.Adapter<SimpleRecyclerAdapter.ViewHolder>
```

# ViewHolder

- Настройка одной ячейки



```
public void bindView(View view, int position) {  
    TextView name = view.findViewById(R.id.name);  
    TextView additional = view.findViewById(R.id.additional);  
    ImageView image = view.findViewById(R.id.image);  
    ...  
}
```

# ViewHolder

```
class ViewHolder extends RecyclerView.ViewHolder {  
    final TextView firstLine;  
    final TextView secondLine;  
  
    public ViewHolder(View itemView) {  
        super(itemView);  
  
        firstLine = (TextView)itemView.findViewById(R.id.fist_line);  
        secondLine = (TextView)itemView.findViewById(R.id.second_line);  
    }  
}
```

# Convert View

- Переиспользование View

```
@Override
public ViewHolder onCreateViewHolder(
    ViewGroup parent, int viewType) {
    return new ViewHolder(
        li.inflate(R.layout.item_list, parent, false));
}
```

# Convert View

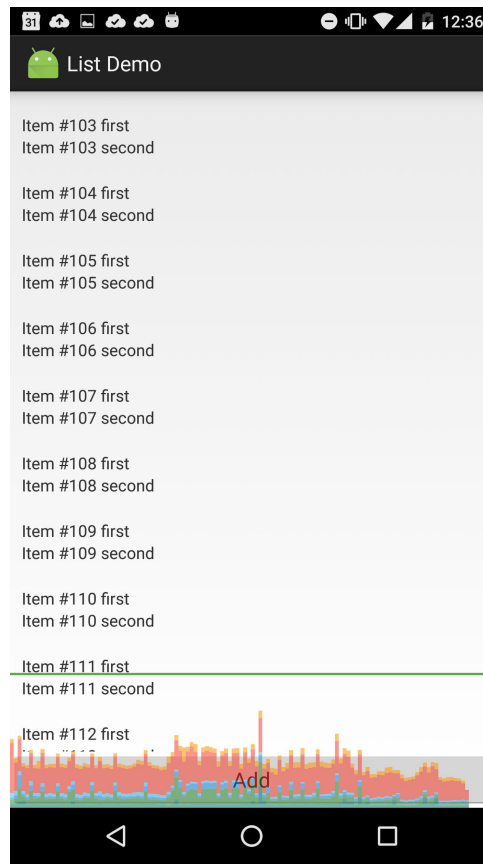
```
@Override  
public void onBindViewHolder(ViewHolder holder, int position) {  
    String str = items.get(position);  
  
    holder.firstLine.setText(str + " first");  
    holder.secondLine.setText(str + " second");  
}
```

# Обработка нажатий

```
public ViewHolder(View itemView) {  
    super(itemView);  
    // ...  
    itemView.setOnClickListener(clickListener);  
}  
  
@Override  
public void onClick(View v) {  
    ViewHolder holder =  
        (ViewHolder)recyclerView.getChildViewHolder(v);  
    int position = recyclerView.getAdapterPositionFor(holder);  
}
```



# Сравнение



Собственные View

# View

- `onDraw(Canvas c)`
  - `drawLine`, `drawRect`, `drawArc`
  - `Paint`
- `onTouchEvent`
  - Возвращем `true`, если обработали
  - `MotionEvent`
    - `getAction`
    - `getX`, `getY`

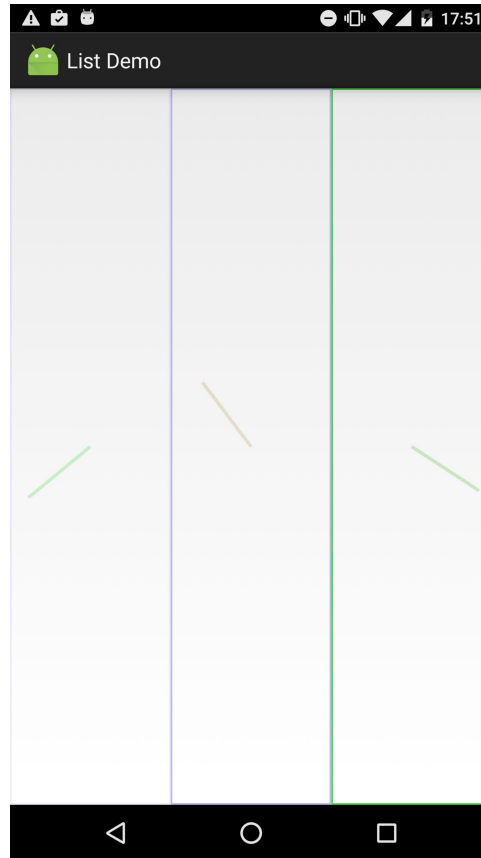
# ViewGroup

- `onMeasure(int widthSpec, int heightSpec)`
  - Верхние 2 бита - mode
    - EXACTLY, AT\_MOST, UNSPECIFIED
  - Остальные - size
  - `setMeasuredDimensions`
- `onLayout(left, top, right, bottom)`
- `View#measure` - измеряем “детей”
  - `measureChild - LayoutParams -> MeasureSpec`

# Зачем это нужно?

- Увеличение производительности
  - Упрощение вёрстки
  - Уменьшение вложенности
- Не хватает возможностей стандартных компонент

# Пример



# Пример. ViewGroup.

```
@Override
protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
    int width = MeasureSpec.getSize(widthMeasureSpec);
    int height = MeasureSpec.getSize(heightMeasureSpec);

    int oneChildWidth = width / childCount;
    int oneChildWidthMeasureSpec = MeasureSpec.makeMeasureSpec(
        oneChildWidth, MeasureSpec.EXACTLY);
    int oneChildHeightMeasureSpec = MeasureSpec.makeMeasureSpec(
        height, MeasureSpec.EXACTLY);

    for (int i = 0; i < childCount; i++) {
        View child = getChildAt(i);

        child.measure(oneChildWidthMeasureSpec, oneChildHeightMeasureSpec);
    }

    setMeasuredDimension(width, height);
}
```

# Пример. ViewGroup.

```
@Override
protected void onLayout(boolean changed, int l, int t, int r, int b) {
    int count = getChildCount();
    int x = 0;
    for (int i = 0; i < count; i++) {
        View childAt = getChildAt(i);

        childAt.layout(x, 0,
            x + childAt.getMeasuredWidth(),
            childAt.getMeasuredHeight());
        x += childAt.getMeasuredWidth();
    }
}
```



# Пример. View.

```
public class AngleView extends View {  
    private float angle;  
  
    private final Paint paintAngle = new Paint();  
    private final Paint paintRect = new Paint();  
  
    public AngleView(Context context, AttributeSet attrs) {  
        super(context, attrs);  
  
        paintAngle.setColor(new Random().nextInt());  
        paintAngle.setStrokeWidth(10);  
  
        paintRect.setColor(new Random().nextInt());  
        paintRect.setStrokeWidth(10);  
        paintRect.setStyle(Paint.Style.STROKE);  
    }  
}
```

# Пример. View.

```
@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);

    ...

    canvas.drawRect(0, 0, width, height, paintRect);

    canvas.drawLine(centerX, centerY,
        (float)(centerX + minSize * Math.cos(angle)),
        (float)(centerY + minSize * Math.sin(angle)), paintAngle);

    angle += 0.1;
    invalidate();
}
```

# Сенсоры

# Сенсоры

```
sensorManager =  
    (SensorManager)context.getSystemService(Context.SENSOR_SERVICE);  
  
sensor = sensorManager.getDefaultSensor(Sensor.TYPE_ORIENTATION);  
  
sensorManager.registerListener(this, sensor,  
    SensorManager.SENSOR_DELAY_NORMAL);  
  
sensorManager.unregisterListener(this);
```

# Сенсоры

```
@Override  
public void onSensorChanged(SensorEvent event) {  
    textView.setText("Value: " + event.values[0]);  
}
```

Вопросы?