Módulo 4 - Interface Gráfica Gerenciadores de Layout

Programação Orientada a Objetos

Prof. Rone Ilídio - UFSJ

Classe View

- Todos os componente gráficos são filhos de View
- Para se criar um novo componente basta estender View e implementar o método onDraw (Canvas) que é responsável por desenhar o componente na tela.
- Dois tipos de componentes:
  - Widgets: filhos de view (Button, ImageView, etc)
  - Layouts: filhos de android.view.ViewGroup (filha de View), são gerenciadores de layout

Gerenciadores de Layout

Principais layouts

- AbsolutLayout: cada componente possuirá uma posição X,Y
- FrameLayout: um componente ocupando a tela inteira
- LinearLayout: componentes organizados na vertical ou horizontal
- TableLayout: organizados em forma de coluna e linhas
- RelativeLayout: um componente relativo a outro

Gerenciadores de Layout

- Para alterar o layout:
  - Clique com o botão direito sobre a tela da aplicação
  - Escolha “Change Layout ...”
  - Escolhe o layout

Gerenciadores de Layout

- Deve-se configurar a largura e a altura do layout principal e de cada componente
- Duas configurações:
  - android:layout_height: altura do componente/View
  - android:layout_width: largura do componente/View
- Esses parâmetros recebem os seguintes valores:
  - Número: número de pixels
  - fill_parent: ocupa todo o espaço
  - Wrap_content: ocupa apenas o espaço necessário

ScrollView

- ScrollView é um gerenciador de layout que permite a criação de barra de rolagem, tanto horizontal como vertical
- Funciona em conjunto com outro gerenciador de layout (que trabalha dentro do ScrollView)
- Veja exemplo

ScrollView

- Crie um novo projeto
  - Escolha uma API 4.1
- Modifique o Layout para ScrollView
- Insira um layout LinearLayout (vertical)
  - Modifique seu nome para linearlayout1
- Insira o seguinte código na ActivityMain

ScrollView

```
public class MainActivity extends Activity {
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      LinearLayout layout = (LinearLayout)findViewById(R.id.linearlayout1);
      for(int i=0; i<100; i++){
         TextView text = new TextView(this);
         text.setLayoutParams( new   LayoutParams(LayoutParams.WRAP_CONTENT,
                                    LayoutParams.WRAP_CONTENT));
         text.setText("Texto:"+i);
         layout.addView(text);
      }
   }
}
```

ScrollView

- Obtém o layout:

```
LinearLayout layout = (LinearLayout)
                    findViewById(R.id.linearlayout1);
```

- *Cria um TextView e insere no layout*

```
TextView text = new TextView(this);
text.setLayoutParams(new  LayoutParams(LayoutParams.
            WRAP_CONTENT, LayoutParams.WRAP_CONTENT));
```

- Insere o textview no layout

```
layout.addView(text);
```

Resultado


10:48
MainActivity
Texto:76
Texto:77
Texto:78
Texto:79
Texto:80
Texto:81
Texto:82
Texto:83
Texto:84
Texto:85
Texto:86
Texto:87
Texto:88
Texto:89
Texto:90
Texto:91
Texto:92
Texto:93
Texto:94
Texto:95
Texto:96
Texto:97
Texto:98
Texto:99

GridView

- Utilizado para visualizações em forma de tabela (linhas e colunas)
- Utilização clássica: álbum de fotos.
- Parâmetros importantes:
  - **android:numColumns**: número de colunas; auto_fit define o número de colunas automaticamente, de acordo com a largura da coluna
  - **android:columnWidth="200px"**: largura de cada coluna

GridView

- Para exibir imagens é necessário criar um ListAdapter com as imagens necessárias
  - Para criá-lo deve-se criar um classe filha da classe abastrata android.widget.BaseAdapter
  - Deve-se implementar seus métodos, de forma que o método getView(int posicao, View v, ViewGroup parent) retorna um ImageView

- Exemplo: exibe uma tabela com imagens
  - As imagens devem estar na pasta \res\drawable-mdpi
- Crie um novo projeto
- Crie a classe filha de BaseAdapter

GridView

- A interface gráfica deve ter um componente do tipo GridView (guia Composite)
- Configure o GridView da seguinte forma:

```
<GridView
android:id="@+id/gridView1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:columnWidth="200px"
android:gravity="center"
android:numColumns="auto_fit" >
```

```
public class AdaptadorImagem extends BaseAdapter{
   private Context ctx;
   private final int[] images;
   private final LayoutParams params;
        public AdaptadorImagem(Context c, int[] imagens, LayoutParams params){
                ctx = c;
                images = imagens;
                this.params = params;
        }
        public int getCount() {
                return images.length;
        }
        public Object getItem(int arg0) {
                return null;
        }
        public long getItemId(int arg0) {
                return 0;
        }
        public View getView(int posicao, View convertView, ViewGroup parent) {
                ImageView img = new ImageView(ctx);
                img.setImageResource(images[posicao]);
                img.setAdjustViewBounds(true);
                img.setLayoutParams(params);
                return img;
        }
}
```

```
public class MainActivity extends Activity implements OnItemClickListener{
    private int[] imagens = {R.drawable.testefigura,R.drawable.ic_launcher,};
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        GridView grid = (GridView)findViewById(R.id.gridView1);
        grid.setAdapter(new AdaptadorImagem(this, imagens,
                                            new GridView.LayoutParams(30,30)));
        grid.setOnItemClickListener(this);
    }
    public void onItemClick(AdapterView av, View v, int posicao, long id) {
        Toast.makeText(MainActivity.this, "Imagem selecionada:" + posicao,
                                                    Toast.LENGTH_LONG).show();
    }
}
```

Gallery

- Semelhante ao GridView

Aplicação com várias guias

Utilizando o TabHost

Utilizando Várias Guias

- Criação de uma tela contendo várias guias
- Activity principal: TabActivity
- O xml associado à activity principal possui um layout do tipo TabHost
- Cada guia é uma activity, com seu xml próprio
- A activity principal faz a união de todas as activities
- Passagem de informações entre activities por meio de variáveis estáticas

Crie a Classe Dados

- Essa classe possuirá somente variáveis estáticas

```
public class Dados {
    public static String nome;
    public static String email;
    public static String curso;
    public static String periodo;
}
```

activity_main.xml

```
<TabHost xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@android:id/tabhost"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

        <TabWidget
            android:id="@android:id/tabs"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >
        </TabWidget>

        <FrameLayout
            android:id="@android:id/tabcontent"
            android:layout_width="match_parent"
            android:layout_height="match_parent" >
        </FrameLayout>
</TabHost>
```

activity_main.xml

- TabHost
  - Layout próprio para esse tipo de aplicação com guias
  - Deve ter: TabWidget e FrameLayout

- TabWidget → representa as abas de cada guia
- FrameLayout → Layout que receberá as activities

ActivityMain.java

```
public class MainActivity extends TabActivity implements OnTabChangeListener {
	@Override
	public void onCreate(Bundle icicle) {
		super.onCreate(icicle);
		// getTabHost() é da TabActivity
		TabHost tabHost = getTabHost();
		tabHost.setup();
		tabHost.setOnTabChangedListener(this);
		// Tab 1 (abrir com Intent -> Activity Tab1.class)
		TabSpec tab1 = tabHost.newTabSpec("Id1");    //Identificador da tab
		tab1.setIndicator("Guia 1", getResources().getDrawable(R.drawable.circulo));
		tab1.setContent(new Intent(this, Tab1.class));
		tabHost.addTab(tab1);
		// Tab 2
		TabSpec tab2 = tabHost.newTabSpec("Id2");     //Identificador da tab
		tab2.setIndicator("Guia 2", getResources().getDrawable(R.drawable.triangulo));
		tab2.setContent(new Intent(this, Tab2.class));
		tabHost.addTab(tab2);
		//tabHost.setCurrentTabByTag("Id1");
	}
	public void onTabChanged(String tabId) {
		Toast.makeText(this, "Tab:"+tabId, Toast.LENGTH_LONG).show();
	}
}
```

ActivityMain.java

- TabActivity → extends Activity
- TabHost
  - Componente formado por várias guias
  - tabHost.setup(): prepara para receber as guias
  - addTab(): adiciona uma nova tab ao tabhost
- TabSpec → uma guia
  - setIndicator: recebe o título e uma figura;
  - setContent: recebe uma intent com a activity a ser aberta.
- OnTabChangeListener
  - Trata a mudança de tab
- Importante: colocar as figuras circulo.png e triangulo.png na pasta res/drawable-mdpi

activity_tab1.xml

- Crie a seguinte interface gráfica
- Utilize os nomes:
  - etNome: 1º EditText
  - etEmail: 2º EditText

Tab1.java

```
public class Tab1 extends Activity {
    @Override
    public void onCreate(Bundle icicle)
    {
        super.onCreate(icicle);
        setContentView(R.layout.activity_tab1);
    }
    @Override
    protected void onResume() {
        super.onResume();
        EditText ednome = (EditText)findViewById(R.id.etNome);
        ednome.setText(Dados.nome);
        EditText edemail = (EditText)findViewById(R.id.etEmail);
        edemail.setText(Dados.email);
    }
```

Continua ...

Tab1.java

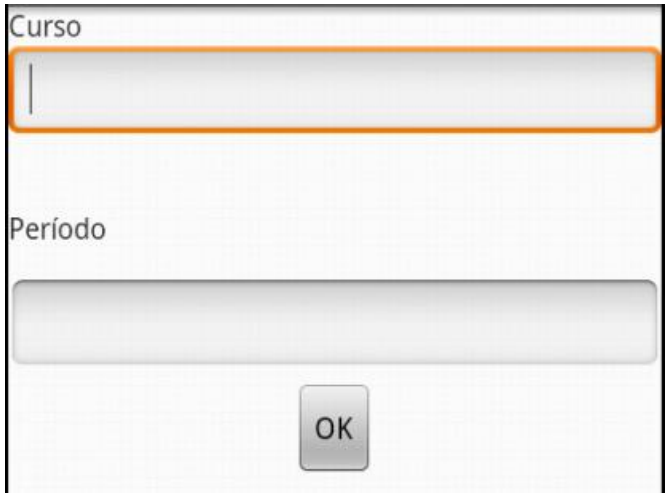
```
    @Override
    protected void onPause() {
        super.onPause();
        EditText ednome = (EditText)findViewById(R.id.etNome);
        Dados.nome = ednome.getText().toString();
        EditText edemail = (EditText)findViewById(R.id.etEmail);
        Dados.email = edemail.getText().toString();
    }
}
```

Tab1.java

- onPause()
  - Executado quando a guia perde o foco
  - Salva as informações da tela em variáveis estáticas
- onResume()
  - Executado quando o foco volta para a guia
  - Pega as informações contidas nas variáveis estáticas e coloca nas caixas de texto

activity_tab2.xml

- Crie a seguinte interface gráfica
- Utilize os nomes:
  - etCurso: 1º EditText
  - etPeriodo: 2º EditText



Tab2.java

```
public class Tab2 extends Activity implements OnClickListener{
    @Override
    public void onCreate(Bundle icicle){
        super.onCreate(icicle);
        setContentView(R.layout.activity_tab2);
        Button btnOk = (Button)findViewById(R.id.button1);
        btnOk.setOnClickListener(this);
    }
    @Override
        protected void onResume() {
        super.onResume();
        EditText edcurso = (EditText)findViewById(R.id.etCurso);
        edcurso.setText(Dados.curso);
        EditText edperiodo= (EditText)findViewById(R.id.etPeriodo);
        edperiodo.setText(Dados.periodo);
```

Continua ...

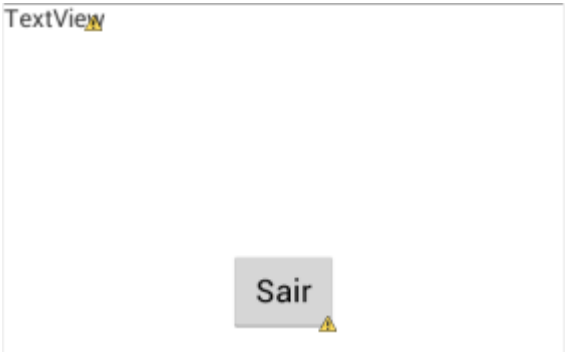
```
    }
```

Tab2.java

```
    @Override
    protected void onPause() {
        super.onPause();
        EditText edcurso = (EditText)findViewById(R.id.etCurso);
        Dados.curso = edcurso.getText().toString();
        EditText edperiodo = (EditText)findViewById(R.id.etPeriodo);
        Dados.periodo = edperiodo.getText().toString();
    }
    @Override
    public void onClick(View v) {
        EditText edcurso = (EditText)findViewById(R.id.etCurso);
        Dados.curso = edcurso.getText().toString();
        EditText edperiodo = (EditText)findViewById(R.id.etPeriodo);
        Dados.periodo = edperiodo.getText().toString();
        Intent it = new Intent(v.getContext(),ResultadoActivity.class);
        startActivity(it);
        finish();
    }
}
```

activity_resultado.xml

- Possui somente um TextView e um botão



ResultadoActivity.java

```
public class ResultadoActivity extends Activity implements OnClickListener {
    public void onCreate(Bundle iCicle){
        super.onCreate(iCicle);
        setContentView(R.layout.activity_resultado);
        TextView tvdado = (TextView)findViewById(R.id.tvResultado);
        tvdado.setText(Dados.nome+" "+Dados.email+" "+Dados.curso+"
            "+Dados.periodo);
        Button btn = (Button)findViewById(R.id.button1);
        btn.setOnClickListener(this);
    }

    @Override
        public void onClick(View v) {
        finish();
    }
}
```