Módulo 2 - Novas Activities Android

Programação Orientada a Objetos

Prof. Rone Ilídio - UFSJ

Inserindo a segunda activity

- Para inserir uma nova activity basta clicar com o botão direito sobre o pacote, novo, outros, Activity.
- Essa operação corresponde a:
  - Inserir uma nova classe no pacote
  - Inserir um novo arquivo xml no em res\layout
  - Associar o xml à nova classe
    - Dentro do método onCreate da nova classe inserir: setContentView(R.layout.*nome_do_xml);*
  - Modificar o arquivo AndroidManifest.xml acrescentando uma tag para a nova activity, ex:

```
<activity android:name=".NomeActivity" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Inserindo a segunda activity

- AndroidManifest.xml
  - <activity android:name=".NomeActivity" >
    ... Definições da activity ...
    <activity>
  - NomeActivity = nome da classe
  - <intent-filter>
    ... Definições de execução
    <\intent-filter>
  - <action android:name="android.intent.action.MAIN" />: activity pode ser inicializada individualmente
  - <category android:name="android.intent.category.LAUNCHER" />: activity pode ser um icone e aparecer junto com as outras aplicações

Chamando outra activity

- Dois métodos
  - startActivity
  - startActivityForResult
- startActivity (intent)
  - Inicia a Activity sem vínculo
- StartActivityForResult(intent, codigo)
  - Inicia uma Activity e espera por um retorno
  - Deve-se dar um código para cada chamada

Exemplo 1– startActivity

- Crie as interfaces gráficas
- Primeira Activity

Segunda Activity

Exemplo 1– startActivity

- Primeira Activity

```
public class MainActivity extends Activity {
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      Handler obj = new Handler();
      Button btn = (Button) findViewById(R.id.button1);
      btn.setOnClickListener(obj);
   }
   public class Handler implements OnClickListener{
        public void onClick(View v){
          if(v.getId() == R.id.button1){
                Intent intent = new Intent(v.getContext(),SegundaActivity.class);
                startActivity(intent);
             }
      }
   }
}
```

Exemplo 1– startActivity

- O objeto da classe Intent será explicado posteriormente
  - Ele é o núcleo do sistema Android
  - Todas as execuções dependem dessa classe
- Intent intent;

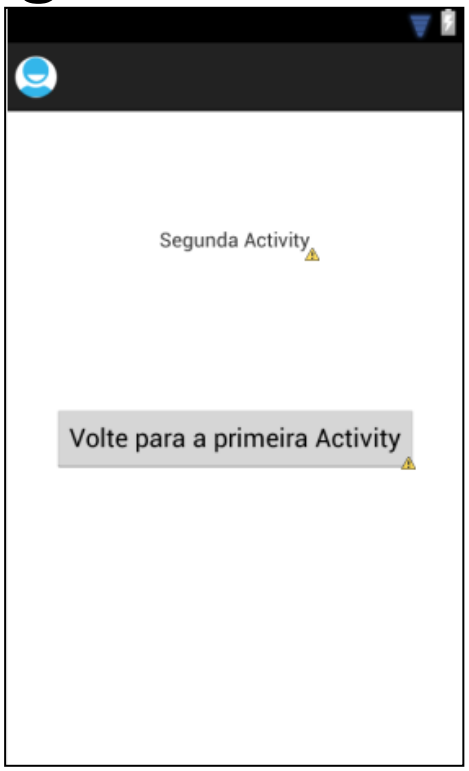
  intent = new Intent(v.getContext(),SegundaActivity.class);
- SegundaActivity.class: nome da Activity que será executada

Exemplo 2– startActivityForResult

- Crie as interfaces gráficas
- Primeira Activity

MainActivity.class e main_activity.xml

Segunda Activity



SegundaActivity.class e activity_segunda.xml

- **1ª Activity**

```
public class MainActivity extends Activity {
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      Handler obj = new Handler();
      Button btn = (Button) findViewById(R.id.button1);
      btn.setOnClickListener(obj);
   }
   public void onActivityResult(int requestCode, int resultCode, Intent data) {
     CharSequence x = data.getCharSequenceExtra("name");
     Toast.makeText(this, x + ", requestCode:"+requestCode+ " resultCode:"+resultCode,
                                                          Toast.LENGTH_LONG).show();
   }
   public class Handler implements OnClickListener{
    public void onClick(View v){
     if(v.getId() == R.id.button1){
       Intent intent = new Intent(v.getContext(),SegundaActivity.class);
       startActivityForResult(intent,0);
     }
    }
   }
}
```

2ª Activity

```
public class SegundaActivity extends Activity {
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_segunda);
      Handler obj=new Handler();
      Button btn = (Button)findViewById(R.id.btnSegunda);
      btn.setOnClickListener(obj);
   }
   public class Handler implements OnClickListener{
     public void onClick(View v){
      if(v.getId() == R.id.btnSegunda){
        Intent intent = new Intent();
        intent.putExtra("name", "ok");
        setResult(10, intent);
        finish();
      }
     }
   }
}
```

Exemplo 2– startActivityForResult

- startActivityForResult(intent,0): zero é o código de requisição da activity
- onActivityResult: método chamado após a conclusão da segunda Activity
  - » requestCode: código de requisição (no caso 0)
  - » resultCode: código de retorno (no caso 10)
  - » Intent data: intent para execução da primeira activity
- Toast.makeText: equivalente ao JOptionPane
  - » Toast.makeText(this, “string”, Toast.LENGTH_LONG).show();
    - String: dado exibido
    - Toast.LENGTH_LONG: tempo de exibição

Informações entre Activities

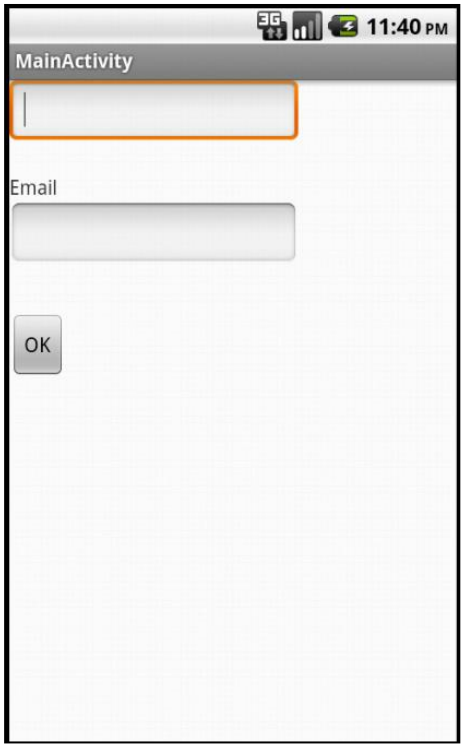
- Uma forma simples de compartilhar dados entre duas ou mais Activities é utilizando o padrão Singleton
- Como exemplo, crie a seguinte classe:

```
public class Dados {
   public static Vector v;
   public static Vector getInstance(){
      if(v == null) v = new Vector ();
      return v;
   }
}
```

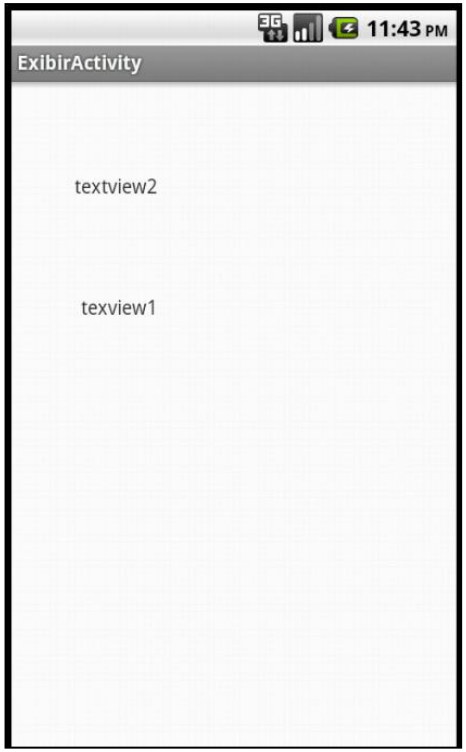
- Todas as Activities terão acesso ao vetor da seguinte forma:
  - Vector a = Dados.getInstance();

Exemplo - Informações entre Activities

- Crie as seguintes interfaces:



MainActivity.java e activity_main.xml



ExibirActivity.java e activity_Exibir.xml

Exemplo - Informações entre Activities

- Crie a classe Dado.java

```
public class Dado {
    public static String nome;
    public static String email;
}
```

Observe que os atributos são static

MainActivity.java

```
public class MainActivity extends Activity  implements OnClickListener{
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      Button btn = (Button) findViewById(R.id.button1);
      btn.setOnClickListener(this);
   }
   public void onClick(View v) {
      EditText tvnome = (EditText)findViewById(R.id.editText1);
      EditText tvemail = (EditText)findViewById(R.id.editText2);
      Dado.nome = tvnome.getText().toString();
      Dado.email = tvemail.getText().toString();
      Intent intent = new Intent(v.getContext(),ExibirActivity.class);
      startActivity(intent);
   }
}
```

ExibirActivity.java

```
public class ExibirActivity extends Activity{
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_exibir);

      TextView tv1 = (TextView)findViewById(R.id.textView1);
      tv1.setText(Dado.email);

      TextView tv2 = (TextView)findViewById(R.id.textView2);
      tv2.setText(Dado.nome);
   }
}
```

Informações entre activities usando Bundle

- Objetos da classe Bundle são utilizados por Intents para passar informações entre Activities
- Bundle possui métodos para cada tipo de dados, exemplo:
  - putString(key,String) → onde key identifica a string
  - putDouble(key,double)
  - putInt(key,int)
- O método putExtras(Bundle) da classe Intent recebe o Bundle que será “enviado” para outra Activity

Informações entre activities usando Bundle

- Para enviar os dados para a segunda Activity

```
Intent intent = new Intent(v.getContext(),ExibirActivity.class);
Bundle params = new Bundle();
params.putString(“chave1", etnome.getText().toString());
intent.putExtras(params);
startActivity(intent);
```

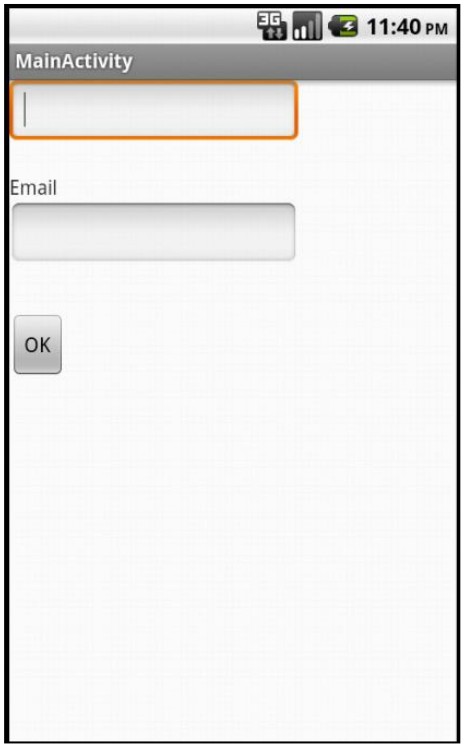
Informações entre activities usando Bundle

- Para receber dentro da segunda Activity

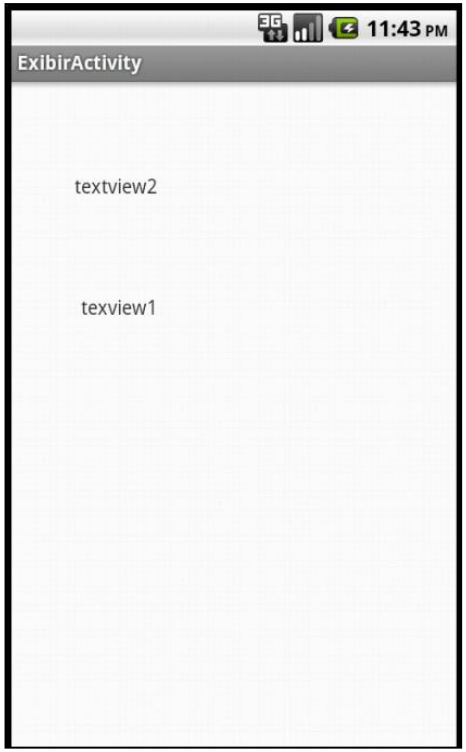
```
String str=“”;
Intent it = getIntent();
 if(it != null){
      Bundle params = it.getExtras();
       if(params != null){
            str = params.getString(“chave");
       }
  }
```

Informações entre activities usando Bundle

- Crie as seguintes interfaces:



MainActivity.java e activity_main.xml



ExibirActivity.java e activity_Exibir.xml

```
public class MainActivity extends Activity implements OnClickListener{
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      Button btn = (Button)findViewById(R.id.button1);
      btn.setOnClickListener(this);
   }
   public void onClick(View v){
      if(v.getId()==R.id.button1){
         EditText etnome = (EditText)findViewById(R.id.editText1);
         EditText etemail = (EditText)findViewById(R.id.editText2);
         Intent intent = new Intent(v.getContext(),ExibirActivity.class);
         Bundle params = new Bundle();
          params.putString("nome", etnome.getText().toString());
         params.putString("email", etemail.getText().toString());
         intent.putExtras(params);
         startActivity(intent);
      }
   }
}
```

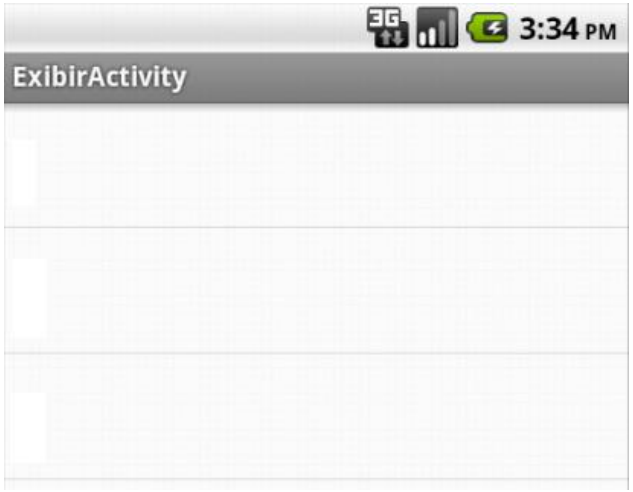
```
public class ExibirActivity extends Activity {
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_exibir);
      String nome="";
      String email="";
      Intent it = getIntent();
      if(it != null){
         Bundle params = it.getExtras();
         if(params != null){
            nome = params.getString("nome");
            email = params.getString("email");
         }
      }
      TextView tvnome = (TextView)findViewById(R.id.textView1);
      tvnome.setText(nome);
      TextView tvemail = (TextView) findViewById(R.id.textView2);
      tvemail.setText(email);
   }
}
```

ListActivity

- ListActivity: filha de Activity
- Criada para manipular uma tela que contenha uma ListView
- A ListView deve-se chamar *@android:id/list*
- A ListView recebe um ArrayAdapter para exibir uma lista contendo seus elementos

ListActivity - Exemplo

- Crie a interface



MainActivity.java e activity_main.xml

ExibirActivity.java e activity_exibir.xml
Obs: o nome do listview deve ser
android:id="*@android:id/list*“
Modifique isso no xml

```
public class MainActivity extends Activity implements OnClickListener{
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      Button btnCadastro = (Button)findViewById(R.id.button1);
      btnCadastro.setOnClickListener(this);
      Button btnExibir = (Button)findViewById(R.id.button2);
      btnExibir.setOnClickListener(this);
   }
   public void onClick(View v){
    if(v.getId() == R.id.button1){
      EditText e = (EditText)findViewById(R.id.editText1);
      Dados.v.add(e.getText().toString());
    }
    if(v.getId() == R.id.button2){
      Intent intent = new Intent(v.getContext(),ExibirActivity.class);
      startActivityForResult(intent, 0);
    }
   }
}
```

```
public class ExibirActivity extends ListActivity {
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_exibir);
      String v[] = new String[Dados.v.size()];
      for(int i=0;i<Dados.v.size();i++){
        v[i] = Dados.v.get(i);
      }
      ArrayAdapter<String> meuarray = new
                  ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,v);
      setListAdapter(meuarray);
   }
   public void onListItemClick(ListView l, View v, int position, long id){
     String text = this.getListAdapter().getItem(position).toString();
     Toast.makeText(this, text, Toast.LENGTH_LONG);
   }
}
```

ListActivity

- Os dados a serem inseridos na lista deve estar dentro de um ArrayAdapter
- **R.layout.*simple_list_item_1:*** define uma lista simples
- O método **onListItemClick** é chamado ao clicar um item da lista
- O métodos setListAdapter é implementado em ListActivity, ele insere um ArrayAdapter na lista

Método finish()

- O Android possui uma pilha de Activities em execução (activity stack)
- A Activity do topo está sendo exibida na tela
- O método startActivity() coloca outra Activity no topo
- O método finish() destrói a Activity do topo e a segunda passa a ser executada.

Exibindo a lista de contatos do telefone

- Primeiramente insira contatos na lista do emulador:
  - Entre no aplicativo Contacts
  - Clique em MENU
  - New Contact
  - Insira alguns contatos

Exibindo a lista de contatos do telefone

- Modifique o AndroidManifest.xml
  - Insira a permissão de utilização da lista de contados
- Para todo recurso utilizado pela aplicação (como GPS, câmera, acesso à internet) o AndroidManifest deve conter a permissão.
- Veja código do AndroidManifest.xml: destaque para a permissão

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="rone.contatos"
   android:versionCode="1"
   android:versionName="1.0" >
   <uses-permission android:name="android.permission.READ_CONTACTS" />
   <uses-sdk
      android:minSdkVersion="8"
      android:targetSdkVersion="15" />
   <application
      android:icon="@drawable/ic_launcher"
      android:label="@string/app_name"
      android:theme="@style/AppTheme" >
      <activity
         android:name=".MainActivity"
         android:label="@string/title_activity_main" >
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
      </activity>
   </application>
</manifest>
```

Exibindo a lista de contatos do telefone

- Para ler os contatos

<uses-permission android:name =
"android.permission.READ_CONTACTS" />

- Para alterar/excluir/inserir contatos;

<uses-permission android:name =
"android.permission.WRITE_CONTACTS" />

Exibindo a lista de contatos do telefone

- Crie uma nova aplicação
  - MainActivity.java
  - activity_main.xml
- Insira um ListView no layout (activity_main.xml)
  - Mude o nome do ListView para *@android:id/list*
- Veja o código da ActivityMain.java

```
public class MainActivity extends ListActivity {
   private ListAdapter adaptador;
   @SuppressWarnings("deprecation")
   @Override
   public void onCreate(Bundle icicle) {
      super.onCreate(icicle);
      Uri uri = ContactsContract.Contacts.CONTENT_URI;
      Cursor c = getContentResolver().query(uri, null, null, null, null);
      startManagingCursor(c);
      String[] colunas = new String[]{ContactsContract.Contacts.DISPLAY_NAME};
      int campos[] = new int[]{R.id.nome};
      adaptador = new SimpleCursorAdapter(this,R.layout.activity_main,c,colunas,campos);
      setListAdapter(adaptador);
   }
   protected void onListItemClick(ListView l, View v, int position, long id){
      super.onListItemClick(l, v, position, id);
      Cursor c= (Cursor) adaptador.getItem(position);
      String nomeColuna = ContactsContract.Contacts.DISPLAY_NAME;
      String nome = c.getString(c.getColumnIndexOrThrow(nomeColuna));
      Toast.makeText(this, "contato selecionado: " + nome, Toast.LENGTH_LONG).show();
   }
}
```