Slide 9.15: Dynamic checkboxes: Java source code
Slide 9.17: Dynamic checkboxes: Java source code (cont.)
Home

Dynamic Checkboxes: Java Code (Cont.)

Line 34: CheckBox checkBox1 = new CheckBox(
Line 34:   getApplicationContext( ));

The method `getApplicationContext` returns the context of the single, global Application object of the current process, whereas `this` is usually associated with to the activity and can be destroyed if the activity is destroyed—there may be multiple activities (more than likely) with a single application.

Line 35: checkBox1.setID(Integer.parseInt(
Line 35:   number.getText( ).toString( )));

The method `setID` sets the identifier for this view. The identifier does not have to be unique in this view's hierarchy. The identifier should be a positive number.

Line 36: checkBox1.setText(title.getText( ).toString( ));

The method `setText` sets the string value of the view.

Line 37: checkBox1.setTextColor(Color.BLACK);

The method `setTextColor` sets the text color for all the states (normal, selected, focused) to be this color. The `Color` class defines methods for creating and converting color ints. Colors are represented as packed ints, made up of 4 bytes: alpha, red, green, blue. For example, the black color is the constant value: -16777216 (0xFF000000).

Line 39: linearBox.addView(checkBox1);

The method `addView` adds a child view. If no layout parameters are already set on the child, the default parameters for this ViewGroup are set on the child.

DynamicCheckbox/app/src/main/java/com/ecs/wenchen/dynamiccheckbox/MainActivity.java

```java
package com.ecs.wenchen.dynamiccheckbox;

import java.util.Iterator;
import java.util.LinkedHashMap;
import java.util.Map;
import java.util.Set;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.EditText;
import android.graphics.Color;

public class MainActivity extends Activity {
  LinearLayout linearBox;
  CheckBox     checkBox, checkBox1;

  @Override
  protected void onCreate( Bundle savedInstanceState ) {
    super.onCreate( savedInstanceState );
    setContentView( R.layout.activity_main );

    linearBox = (LinearLayout) findViewById( R.id.linearBox );
    final EditText number = (EditText) findViewById( R.id.number );
    final EditText title  = (EditText) findViewById( R.id.title );
    final Button   button = (Button)   findViewById( R.id.add );

    button.setOnClickListener( new View.OnClickListener( ) {
      public void onClick( View v ) {
        checkBox1 = new CheckBox( getApplicationContext( ) );
        checkBox1.setId( Integer.parseInt( number.getText( ).toString( ) ) );
        checkBox1.setText( title.getText( ).toString( ) );
        checkBox1.setTextColor( Color.BLACK );
        checkBox1.setOnClickListener( getOnClickDoSomething( checkBox1 ) );
        linearBox.addView( checkBox1 );
      }
    } );

    final LinkedHashMap<String, String>
      alphabet = new LinkedHashMap<String, String>( );
    alphabet.put( "260", ".NET" );
    alphabet.put( "370", "Computer Architecture" );

    Set<?> set = alphabet.entrySet( );
    // Get an iterator.
    Iterator<?> i = set.iterator( );
    // Display elements.
    while ( i.hasNext( ) ) {
      @SuppressWarnings( "rawtypes" )
      Map.Entry me = ( Map.Entry ) i.next( );
      checkBox     = new CheckBox( this );
      checkBox.setId( Integer.parseInt( me.getKey( ).toString( ) ) );
      checkBox.setText( me.getValue( ).toString( ) );
      checkBox.setOnClickListener( getOnClickDoSomething( checkBox ) );
      linearBox.addView( checkBox );
    }
  }

  View.OnClickListener getOnClickDoSomething( final Button button ) {
    return new View.OnClickListener( ) {
      public void onClick( View v ) {
        final TextView tvView = (TextView) findViewById( R.id.textView3 );
        tvView.setText( button.getId( ) + ": " + button.getText( ).toString( ) );
      }
    };
  }
}
```

Slide 9.15: Dynamic checkboxes: Java source code
Slide 9.17: Dynamic checkboxes: Java source code (cont.)
Home