**DatePickerDialog и TimePickerDialog**

По умолчанию в Android уже определены два диалоговых окна - **DatePickerDialog** и **TimePickerDialog**, которые позволяют выбрать дату и время.

Кроме установки даты DatePickerDialog позволяет обработать выбор даты с помощью слушателей**OnDateChangedListener** и **OnDateSetListener**. Что позволяет использовать выбранную дату далее в приложении.

Подобным образом TimePickerDialog позволяет обработать выбор времени с помощью слушателей**OnTimeChangedListener** и **OnTimeSetListener**

Пи работе с данными компонентами надо учитывать, что отсчет месяцев в DatePickerDialog начинается с нуля, то есть январь будет иметь номер 0, а декабрь - 11. И аналогично в TimePickerDialog отсчет секунд и минут будет идти с 0 до 59, а часов - с 0 до 23.

Используем DatePickerDialog и TimePickerDialog в приложении. Определим следующую разметку интерфейса:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | <LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"      xmlns:tools="<http://schemas.android.com/tools>"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:orientation="vertical">        <TextView android:id="@+id/currentDateTime"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:textSize="18dp" />        <Button android:id="@+id/timeButton"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:text="Изменить время"          android:onClick="setTime" />        <Button android:id="@+id/dateButton"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:text="Изменить дату"          android:onClick="setDate"  />    </LinearLayout> |

Здесь определены две кнопки для выбора даты и времени и текстовое поле, отображающее выбранные дату и время. И изменим код activity:

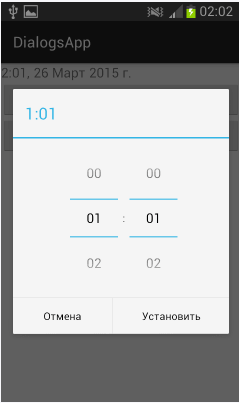
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69 | import android.app.DatePickerDialog;  import android.app.TimePickerDialog;  import android.support.v7.app.ActionBarActivity;  import android.os.Bundle;  import android.text.format.DateUtils;  import android.view.Menu;  import android.view.MenuItem;  import android.view.View;  import android.widget.DatePicker;  import android.widget.TextView;  import android.widget.TimePicker;  import java.util.Calendar;    public class MainActivity extends ActionBarActivity {        TextView currentDateTime;      Calendar dateAndTime=Calendar.getInstance();      @Override      public void onCreate(Bundle savedInstance) {              super.onCreate(savedInstance);              setContentView(R.layout.activity\_main);              currentDateTime=(TextView)findViewById(R.id.currentDateTime);              setInitialDateTime();      }        // отображаем диалоговое окно для выбора даты      public void setDate(View v) {              new DatePickerDialog(MainActivity.this, d,              dateAndTime.get(Calendar.YEAR),              dateAndTime.get(Calendar.MONTH),              dateAndTime.get(Calendar.DAY\_OF\_MONTH))              .show();      }        // отображаем диалоговое окно для выбора времени      public void setTime(View v) {              new TimePickerDialog(MainActivity.this, t,              dateAndTime.get(Calendar.HOUR\_OF\_DAY),              dateAndTime.get(Calendar.MINUTE), true)              .show();      }      // установка начальных даты и времени      private void setInitialDateTime() {            currentDateTime .setText(DateUtils.formatDateTime(this,                          dateAndTime.getTimeInMillis(),                          DateUtils.FORMAT\_SHOW\_DATE | DateUtils.FORMAT\_SHOW\_YEAR                          | DateUtils.FORMAT\_SHOW\_TIME));      }        // установка обработчика выбора времени      TimePickerDialog.OnTimeSetListener t=new TimePickerDialog.OnTimeSetListener() {          public void onTimeSet(TimePicker view, int hourOfDay, int minute) {              dateAndTime.set(Calendar.HOUR\_OF\_DAY, hourOfDay);              dateAndTime.set(Calendar.MINUTE, minute);              setInitialDateTime();          }      };        // установка обработчика выбора даты      DatePickerDialog.OnDateSetListener d=new DatePickerDialog.OnDateSetListener() {          public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {              dateAndTime.set(Calendar.YEAR, year);              dateAndTime.set(Calendar.MONTH, monthOfYear);              dateAndTime.set(Calendar.DAY\_OF\_MONTH, dayOfMonth);              setInitialDateTime();          }      };  } |

Ключевым классом здесь является **java.util.Calendar**, который хранится в стандартной библиоетке классов Java. В методе setInitialDateTime() мы получаем из экземпляра этого класса количество миллисекундdateAndTime.getTimeInMillis() и с помощью форматирования выводим на текстовое поле.

Метод setDate(), вызываемый по нажатию на кнопку, отображает окно для выбора даты. При создании окна его объекту передается обработчик выбора даты DatePickerDialog.OnDateSetListener, который изменяет дату на текстовом поле.

Аналогично метод setTime() отображает окно для выбора времени. Объект окна использует обработчик выбора времени TimePickerDialog.OnTimeSetListener, который изменяет время на текстовом поле.

И поле запуска, нажав на кнопку изменения времени, мы сможем установить время:



Для установки времени на диалоговом окне определена кнопка "Установить". Аналогично работает окно установки даты:

