**Диалоги (справочник)**

*Диалог* - небольшое окошко отображаемое на переднем плане текущей активности.

Андроид предоставляет четыре стандартных диалога:

* ***AlertDialog*** - диалог с от 0 до 3 кнопок или со списком элементов, которые может выбрать пользователь;
* ***ProgressDialog*** - расширение AlertDialog для уведомления пользователя о выполнении какой-то долгосрочной задачи;
* ***DatePickerDialog*** - диалог выбора даты;
* ***TimePickerDialog*** - диалог выбора времени;

**диалоги и активность**

При работе с диалогами рекомендуется следующий подход.

1. Понять какие диалоги нужны в данной активности и пронумеровать их, номер будет служить идентификатором диалога в активности.

static final int DIALOG\_PAUSED\_ID = 0, DIALOG\_GAMEOVER\_ID = 1;

2. Перегрузить в активности метод

protected Dialog onCreateDialog(int id) {

Dialog dialog;

switch(id) {

case DIALOG\_PAUSED\_ID:

// создаем диалог паузы

break;

case DIALOG\_GAMEOVER\_ID:

// создаем диалог конца игры

break;

default:

dialog = null;

}

return dialog;

}

3. Отобразить диалог методом showDialog(номер\_диалога) внутри активности;

4. Завершение диалога делается методом dismiss/cancel в самом диалоге, или методом dismissDialog(номер\_диалога) внутри активности. Если нужно освободить ресурсы связанные с диалогом, его завершают методом removeDialog(int) из активности.

Если диалоги будут использоваться в нескольких активностях, то их удобно вынести в отдельную базовую активность.

**AlertDialog**

Вспомогательный класс *Builder* позволяет создать нужный диалог заранее.

Данный вид диалога может служить для предупреждения или подтверждения выполнения действия, например выход из приложения.

// пример для darkraha.com

// не рекомендуется для андроид приложений, но иногда нужно

public static AlertDialog exitFromApp(final Context context, String resMsg,

String no, String exit, final boolean systemexit) {

AlertDialog.Builder builder = new AlertDialog.Builder(context);

builder.setMessage(resMsg);

builder.setPositiveButton(exit, new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

((Activity) context).finish();

if (systemexit) {

System.exit(0);

}

}

});

builder.setNegativeButton(no, null);

return builder.create();

}

// диалог с двумя кнопками да/нет

public static AlertDialog DlgYesNo(final Context context, String msg,

String yes, String no, DialogInterface.OnClickListener whenYes

) {

if (yes == null) {

yes = "Yes";

}

if (no == null) {

no = "No";

}

AlertDialog.Builder builder = new AlertDialog.Builder(context);

builder.setMessage(msg).setCancelable(false)

.setPositiveButton(yes, whenYes)

.setNegativeButton(no, new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

dialog.cancel();

}

});

return builder.create();

}

AlertDialog также позволяет пользователю выбрать действие из указанного списка.

String op [] = new String [] {"copy","paste",

"delete","upload"};

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setTitle("операции");

DialogInterface.OnClickListener lclick = new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

dialog.dismiss();

switch(which){

case 0:

...

}

}

};

builder.setItems(op,lclick);

dlgOpLocal = builder.create();

**ProgressDialog**

Используется как индикатор прогресса выполнения процесса. Можно добавить кнопку отмены для прерывания процесса.

public static ProgressDialog getWait(final Context context, String message,

String buttonLabel, DialogInterface.OnClickListener onClick,

final AsyncTask t) {

ProgressDialog dlg = new ProgressDialog(context);

dlg.setMessage(message);

dlg.setIndeterminate(true);

dlg.setCancelable(true);

dlg.setProgressStyle(ProgressDialog.STYLE\_SPINNER);

if (onClick == null) {

onClick = new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int which) {

if (t != null) {

t.cancel(true);

}

dialog.dismiss();

return;

}

};

}

// методы setButton, setButton2, setButton3

// считаются устаревшими

dlg.setButton(buttonLabel, onClick);

return dlg;

}

**пользовательский диалог**

Класс Dialog является базовым классом для все диалогов. Ниже приведены некоторые методы:

* ***cancel***() - выбрать отмену диалога;
* ***dismiss***() - завершить диалог и убрать его с экрана;
* ***findViewById***(int id) - найти элемент gui по его идентификатору;
* ***getContext***() - получить контекст в котором запущен диалог;
* ***hide***() - скрыть, но не завершить диалог;
* ***isShowing***() - true, если диалог аотображается;
* ***requestWindowFeature***(int featureId) - запросить дополнительные возможности для окна;
* ***setCancelable***(boolean flag) - true, если диалог может быть отменен клавишей BACK;
* ***setCanceledOnTouchOutside***(boolean cancel) - true, если диалог может быть отменен касанием за пределами окна;
* ***setContentView***(int layoutResID) - установить группу gui элементов по ресурсу (есть варианты метода);
* ***setOnCancelListener***(DialogInterface.OnCancelListener listener) - установить обработчик отмены диалога;
* ***setTitle***(CharSequence title) - установить заголовок окна диалога;
* ***show***() - запустить диалог и отобразить его на экране;

Для создания собственного диалога можно создать объект типа Dialog и настроить его. Данный процесс похож на создание формы для активности.

Context mContext = getApplicationContext();

Dialog dialog = new Dialog(mContext);

dialog.setContentView(R.layout.custom\_dialog);

dialog.setTitle("Custom Dialog");

TextView text = (TextView) dialog.findViewById(R.id.text);

...

Можно создать дочерний класс и настроить диалог изнутри аналогичным образом.

И наконец можно изменить визуальный облик стандартных диалогов.

AlertDialog.Builder builder;

AlertDialog alertDialog;

Context mContext = getApplicationContext();

// парсер xml описаний форм

LayoutInflater inflater = (LayoutInflater)

mContext.getSystemService(LAYOUT\_INFLATER\_SERVICE);

// форма, т.е. группа элементов пользовательского интерфейса

View layout = inflater.inflate(R.layout.custom\_dialog,

(ViewGroup) findViewById(R.id.layout\_root));

TextView text = (TextView) layout.findViewById(R.id.text);

text.setText("не стандартный диалог!");

...

builder = new AlertDialog.Builder(mContext);

builder.setView(layout);

alertDialog = builder.create();

**события**

Кроме стандартных событий типа нажатия клавиши, диалоги имеют два специфических события:

* ***setOnCancelListener***(DialogInterface.OnCancelListener listener) - устанавить обработчик отмены диалога (после вызова метода cancel или нажатии клавиши BACK);
* ***setOnDismissListener***(DialogInterface.OnDismissListener listener) - установить обработчик завершения диалога;