

Лабораторная работа № 1  
Пример разработки пользовательских  
интерфейсов приложения на платформе  
Android

## Цель лабораторной работы

---

Обучение базовым приемам разработки приложений на платформе Android, включающим в себя установку и настройку среды разработки, освоение программы с пользовательским интерфейсом, проведение сеанса отладки.

[www.math.spbu.ru/user/gran/Atom21/lab01](http://www.math.spbu.ru/user/gran/Atom21/lab01)

# План

---

- ▶ Установка и настройка среды разработки
- ▶ Описание приложения
- ▶ Архитектура приложения
- ▶ Модель «Дизайн-XML»
- ▶ Используемые API
- ▶ Установка на устройство
- ▶ Отладка

# Установка и настройка среды разработки

---

- ▶ Установить среда разработки Eclipse
- ▶ <http://www.eclipse.org/downloads/>
- ▶ Установить JDK — Java Development Kit,
- ▶ <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- ▶ Для установки ADT откройте панель расширения среды (Install New Software), и затем в появившемся окне добавьте новый путь - <https://dl-ssl.google.com/android/eclipse/>
- ▶ Далее Eclipse выполнит поиск требуемых данных, и в выпадающем списке вы сможете выбрать ADT и установить его. После установки у вас появится AVD Manager. Его можно запустить прямо из Eclipse (Window-> Android SDK and AVD Manager)
- ▶ Настройте Virtual Device
- ▶ Установите версии API



# Необходимые знания и навыки

---

- ▶ Общие представления об ОС Android
- ▶ Знакомство с материалом лекции № 2
- ▶ Базовое знание языка программирования Java



# Необходимые программные и аппаратные средства

---

- ▶ ПК под ОС Windows (или любой другой, подходящей для Android SDK)
- ▶ Устройство под управлением Android или эмулятор Android SDK



# Обзор приложения

---

- ▶ Расположение элементов управления: TabLayout, LinearLayout, RelativeLayout и GridLayout
- ▶ Простейшие элементы управления: Button, EditText, TextView
- ▶ Список GridView

## Архитектура приложения

---

**Идея.** Обычно, если приложение не имеет сложную структуру, то в одном состоянии приложения используется только один стиль размещения. Мы рассмотрим приложение, которое содержит в себе много различных layouts. Достичь этого, можно путем создания в качестве главного layout, TabLayout, у которого каждый tab имеет свой способ размещения.

### Структура.

- TabLayout →
  - Tab1: -> LinearLayout.
  - Tab2: -> RelativeLayout.
  - Tab3: -> GridLayout.



## Архитектура Tab1

---

Интерфейс для работы с датой в Андроиде

### Компоненты

- TextView - простое текстовое поле, в котором
- будет отображаться текущая или вручную установленная дата.
- Button - кнопка, которая инициирует появление окошка DatePicker, в котором отображается текущая дата.

В окне стандартно есть контролы позволяющие изменить дату.



## Архитектура Tab2

---

### Некоторые виджеты UI

- **Компоненты.**

- TextView - текстовое поле отображающее статическую информацию.
- EditText - текстовое поле с поддержкой редактирования.
- Button - кнопка ОК.
- Button - кнопка Cancel.
- Spinner - выпадающий список с обработкой события выбора.



## Архитектура Tab3

---

Графические объекты (картинки) размещенные в виде таблицы

### Компоненты.

- GridView — представление, отображающее элементы в двумерной сетке с поддержкой вертикального или горизонтального скрола. Элементы, ассоциированные с Grid представлением, должны быть представлены с помощью ListAdapter.



# Дизайн-XML

---

- ▶ Традиционное разделение кода (Java) и ресурсов (XML) — дизайна, строк и т.п.
- ▶ Удобно управлять ресурсами
- ▶ Упрощение локализации
- ▶ Разделение работы программиста и дизайнера



# Используемые API

Date Picker — виджет позволяющий работать с датой.

Создать этот виджет можно при помощи конструктора:

`DatePickerDialog(Context, DatePickerDialog.OnDateSetListener, Year, Month, Day)`.



Установить дату можно при помощи объекта слушателя `DatePickerDialog.OnDateSetListener()` , который вызывается всякий раз, когда пользователь нажимает на кнопку «Set». Реализуя метод экземпляра слушателя `OnDataSet(DataPicker, year, month, day)`, мы сможем получить данные о дате из виджета и в дальнейшем работать с ними.



## Компонеты Tab2

Spinner — это виджет похожий на выпадающий вниз список, отображающий один элемент и позволяющий выбрать любой другой, среди содержимого раскрывающегося списка.

- Spinner в программе можно создать с помощью конструкторов:
  - Spinner(Context, int mode) — параметр mode позволяет задать способ отображения элементов списка `MODE_DIALOG` или `MODE_DROPDOWN`.
  - Spinner(Context context, AttributeSet attrs, int defStyle, int mode).

Элементы спиннера определяются классом `Adapter`, а затем присоединяются к нему с помощью метода `SetAdapter(adapter)`.



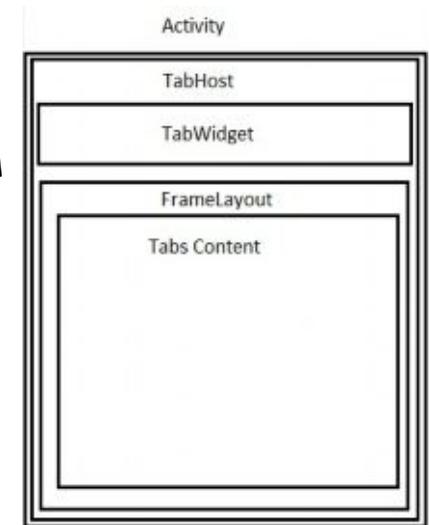
## Tab Layout

Рассмотрение структуры приложения начнем с графических компонентов. В приложении все компоненты UI определяются в xml файле. Основным менеджером размещения в примере является TabLayout.

Использовать менеджер закладок можно двумя подходами:

- Можно создать представления для каждой закладки в одной Activity(форма приложения), а затем прицепить их к TabHost.
- Можно описать представления конкретными Activity и через главный Activity связать их с TabHost.

В приложении используется первый подход (см. Рис.).



## Установка на устройство и отладка

---

Проделайте следующие шаги, чтобы посмотреть предварительно подготовленное приложение в действии.

- ▶ 1). Импорт приложения. Загрузите пример из каталога [lab01 файла labAtom21.rar](#).
- ▶ 2). Запуск приложения.

Run as->Androidapplication

В результате проект перестроится и запустится приложение. После загрузке эмулятора можно убедиться в работоспособности программы и проанализировать ее код по описанию работы.

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ



**Благодарю за внимание!**

**Вопросы?**