

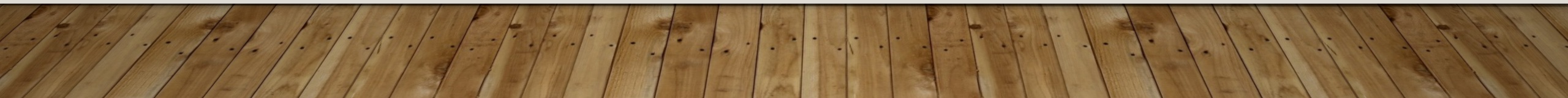
# РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

---

ИСПОЛЬЗОВАНИЕ БИБЛИОТЕК И РАБОТА С БАЗАМИ ДАННЫХ И ГРАФИКОЙ



**Аннотация:** Прежде чем браться за решение какой-то вспомогательной задачи, следует сначала выяснить, не была ли она решена кемто ранее. Повторное использование кода позволяет сберечь ресурсы на выполнение проекта. Такие возможности предоставляют подключаемые библиотеки.



# Библиотеки

## Использование библиотек

**Библиотека (от англ. library)** в программировании - сборник подпрограмм или объектов, используемых для разработки программного обеспечения (ПО). Для ОС **Android** существует большое количество подключаемых библиотек. Их можно классифицировать в зависимости от их предназначения. Выделим следующие группы:



- **Библиотеки совместимости.** Они позволяют использовать возможности, появившиеся в какой-то версии ОС **Android**, на более ранних версиях платформы. Дело в том, что новые версии **API** выходят гораздо быстрее, чем в широком использовании оказываются устройства, поддерживающие эту версию. Разработчик с одной стороны должен ориентироваться на новые возможности и уметь их использовать, а с другой - стараться сделать так, чтобы приложение работало на максимальном количестве устройств. Библиотеки совместимости позволяют сделать это противоречие менее жестким.





- **Библиотеки специального назначения.** Используются для разработки игр, работы с социальными сетями, сбора статистики и в других случаях.
- **Библиотеки, предоставляющие дополнительные возможности.** В эту категорию можно отнести большое количество самых разных библиотек. Сюда можно отнести библиотеки рисования графиков, работы с изображениями, модифицированные элементы управления и многое другое.



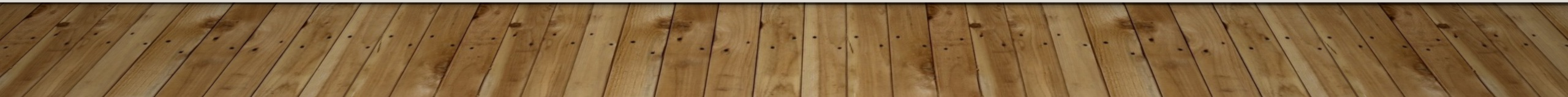
## Подключение библиотек

Библиотеки могут поставляться как в собранном и уже готовом к использованию виде (**jar-файлы**), так и в исходниках. Подключить библиотеку (**файл \*.jar**) очень просто. Достаточно создать папку **libs** в проекте (на том же уровне, что и папки **src** и **res**) и копировать туда файл библиотеки (можно просто перетащить). Далее необходимо добавить ее в проект через меню **Project -> Properties**. Если библиотека представлена в виде исходного кода, необходимо ее предварительно собрать. Необходимо щелкнуть правой кнопкой по **корневой папке проекта - > Export: -> Java - > Runnable JAR file - > Указать класс для запуска -> Указать место сборки -> Finish**

# Обзор популярных библиотек Android Support Library

**Android Support Library** - это набор библиотек, которые обеспечивают обратную совместимость новых **API** на более старых версиях платформы. Каждая библиотека из этого набора обладает обратной совместимостью к конкретному уровню **Android API**. Это означает, что ваши приложения смогут использовать возможности библиотеки и быть запущены на устройствах **Android 1.6 (API level 4)** и выше.

Подключение библиотек поддержки в **Android** является хорошим тоном в разработке приложений, зависящих от версии и возможностей платформы. Использование возможностей **Support Library** поможет вам распространить ваше приложение для большего числа пользователей.





Если вы используете примеры **Android-приложений**, вы можете заметить, что все они содержат по умолчанию одну или несколько библиотек поддержки.

О возможностях различных версий **Android Support Library** можно узнать на официальном сайте. Скачать и установить эти библиотеки можно с помощью **Android SDK Manager**, выбрав в разделе **Extras** нужные пункты.

При настройке обратной совместимости необходимо отредактировать файл манифеста, указав в нем минимальную версию **Android SDK**, которая необходима для запуска приложения, и основную (целевую) версию:

```
<uses-sdk
```

```
android:minSdkVersion="7"
```

```
android:targetSdkVersion="17"/>
```

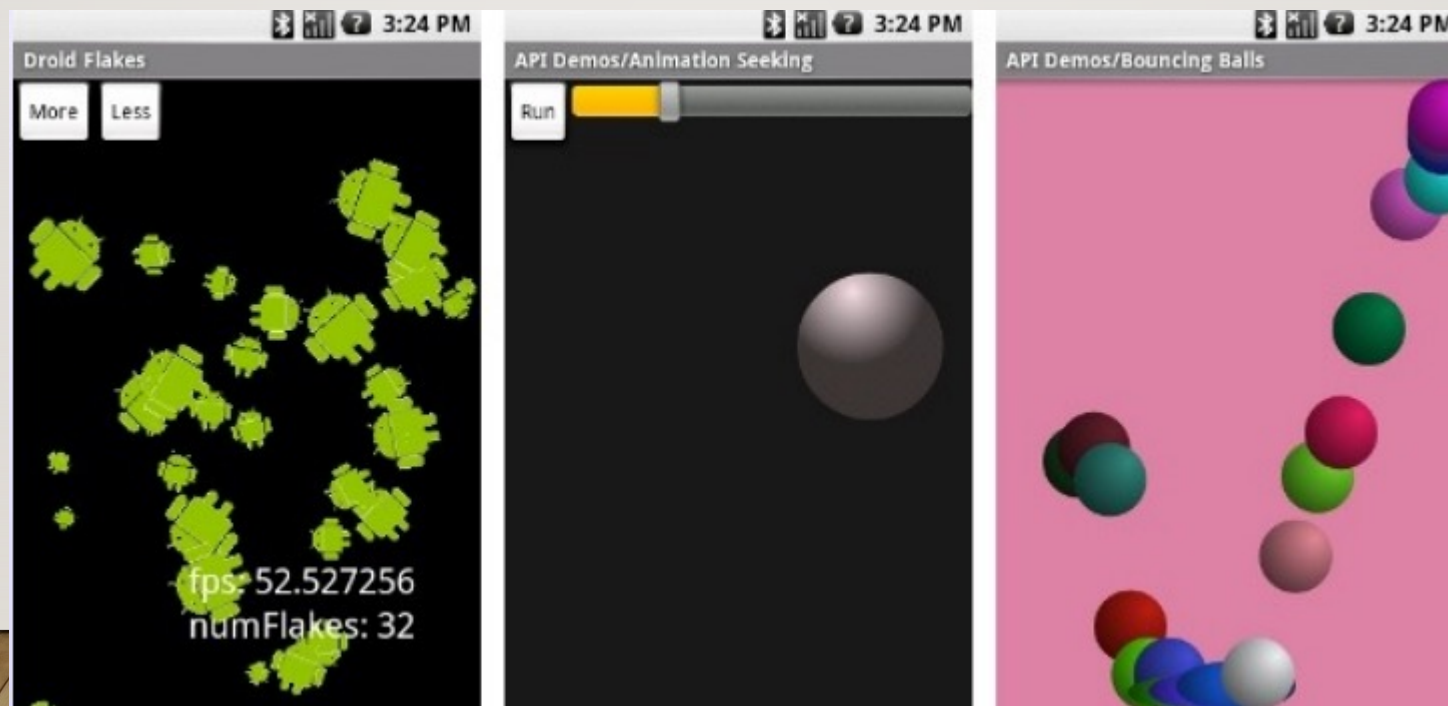


## Сторонние библиотеки

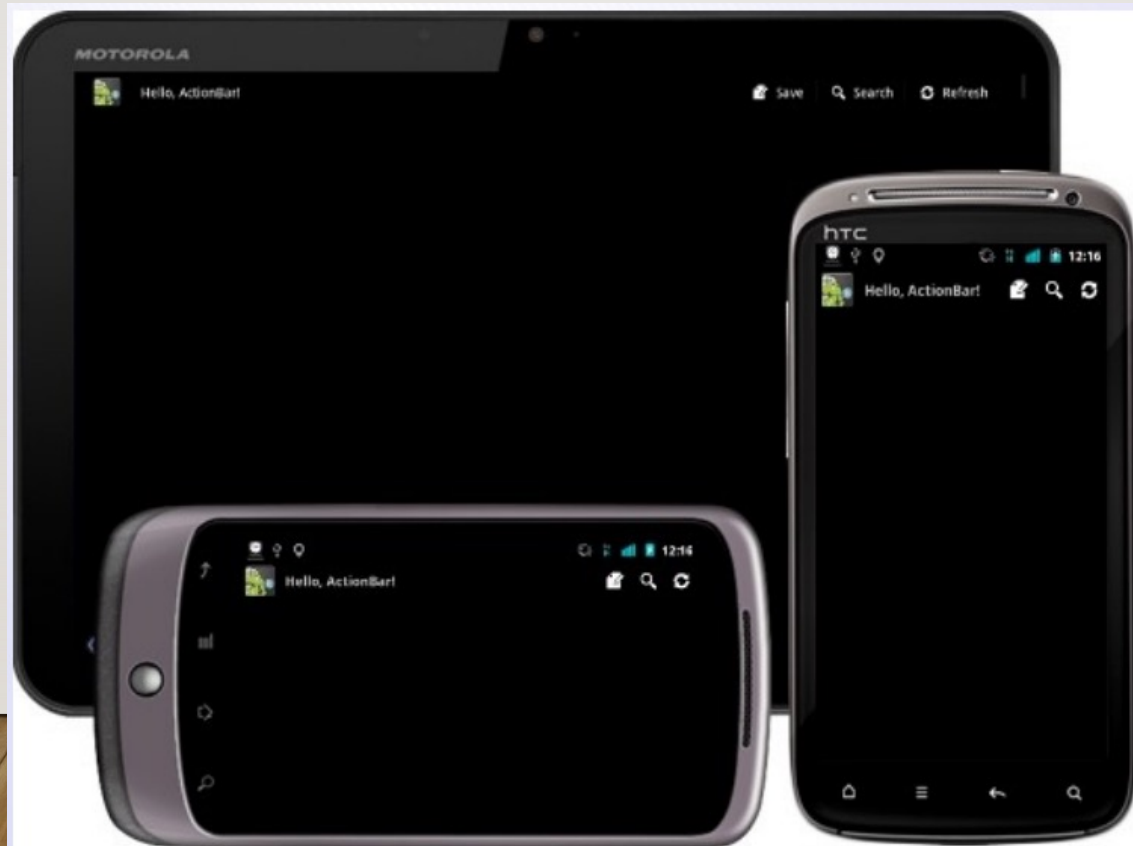
Помимо официальных и поддержанных Google библиотек совместимости, существуют аналогичные решения от сторонних разработчиков. Вы можете их использовать на свой страх и риск, но в некоторых случаях они предпочтительнее, так как обладают некоторыми дополнительными возможностями.



**NineOldAndroids** - один из примеров таких библиотек. Она предназначена для использования анимации, которая стала доступна только с версии **Honeycomb (Android 3.0)**. Она поддерживает различные возможности анимации и очень удобна в использовании. Главным преимуществом этой библиотеки является то, что она работает для всех версий **Android**, начиная с 1.0.



**ActionBarSherlock.** С ее помощью можно использовать нативный компонент **ActionBar**, появившийся только в версии **Android 4.0**, в более ранних (2.x и выше). Ее можно загрузить с официального сайта. Там же содержатся подробные указания по работе с этой библиотекой, имеются примеры. На рис. представлена работа приложения, использующего библиотеку **ActionBarSherlock**, на устройствах со старыми версиями.



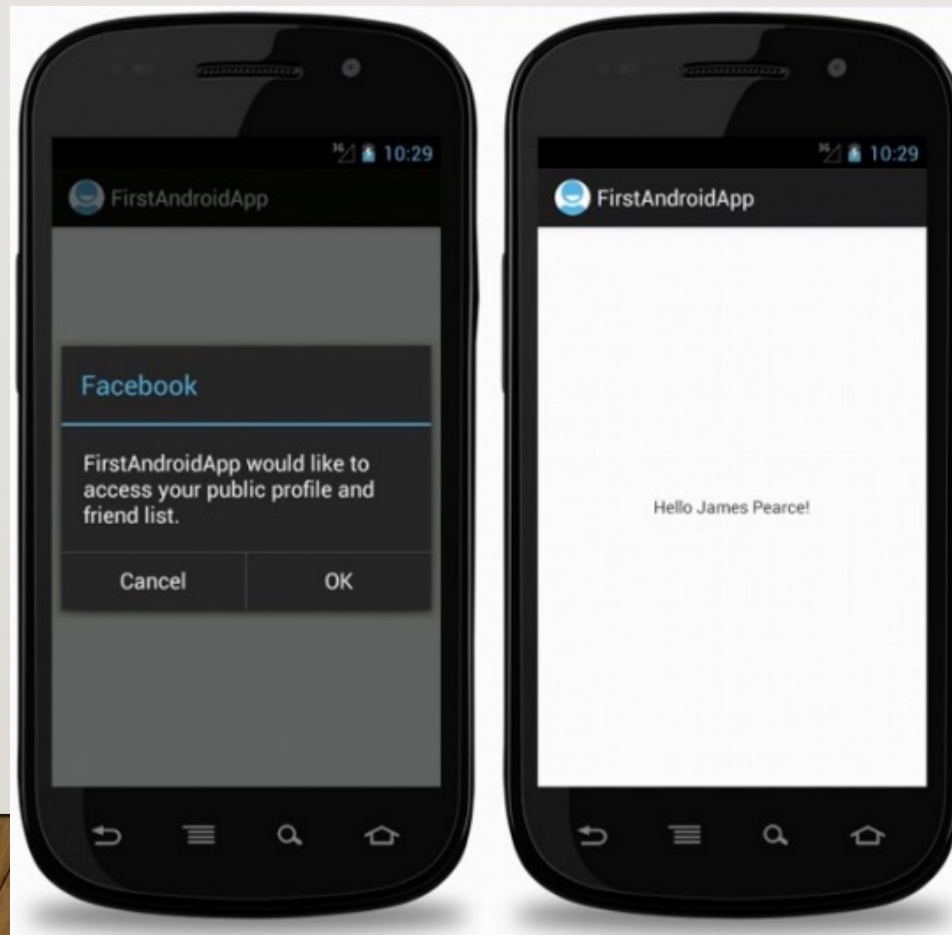


# Библиотеки специального назначения

**Yandex.Metrica for Apps** - набор библиотек для сбора статистики использования мобильного приложения. Метрика показывает актуальную статистику об использовании приложения. сервис позволяет отвечать на вопросы об аудитории и выделять любые ее сегменты. Инструменты помогают понять, как люди пользуются приложением. SDK позволяет отслеживать следующие данные:

- информация об устройстве;
- информация о сессиях;
- информация об источнике перехода пользователя на страницу скачивания приложения;
- действия, выполненные пользователем в приложении;
- местоположение пользователя;
- ошибки, возникающие во время использования приложения;
- собственные события;
- другие данные (например, количество пользователей, установивших приложение).

**Facebook SDK for Android** - официальная библиотека **Facebook** для **Android**. Позволяет писать сообщения на стену, читать и менять статусы, смотреть ленту друзей и многое другое. Официальный сайт содержит большое количество примеров и указаний по разработке приложений



# Прикладные библиотеки

К этой категории можно отнести различные библиотеки, предоставляющие дополнительные возможности.

**Universal Image Loader for Android** - мощная и гибкая библиотека, предназначенная для загрузки, кеширования и отображения картинок в **Android**. Подробности на сайте. Возможности:

- Многопоточная загрузка изображений.
- Широкие возможности настройки и конфигурирования.
- Кеширование загруженных изображений как в оперативной памяти, так и на карте.
- Поддержка виджетов.
- Поддерживает Android 2.0 и выше.





**jsoup: Java HTML Parser** предназначена для парсинга **HTML** страниц. Предоставляет очень удобный API для извлечения данных и манипуляции с ними, используя **DOM**, **CSS** и методы в стиле **jQuery**. Поддерживает спецификации **HTML5** и позволяет парсить страницы так же, как это делают современные браузеры.

Возможности:

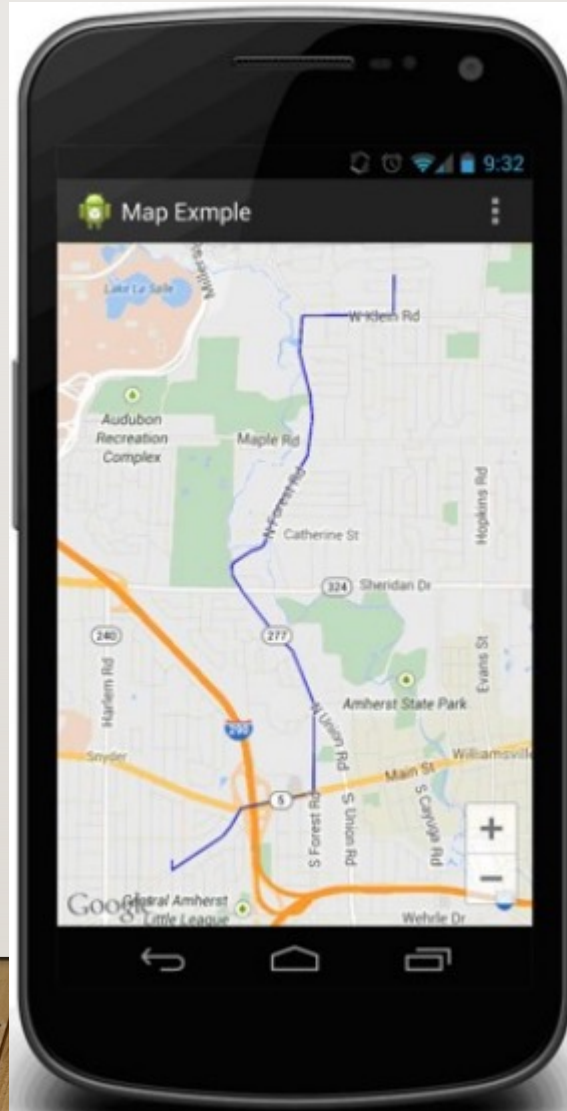
- Может принимать в качестве параметра **URL**, файл или строку.
- Находит и извлекает данные, используя **DOM** и селекторы **CSS**.
- Позволяет манипулировать **HTML-элементами**, атрибутами и текстом.
- Выводит чистый HTML.



**Android Holo ColorPicker** - удобная библиотека, позволяющая выбирать цвет с использованием цветового колеса, выполненная в официально рекомендованном стиле **Holo**. Сайт библиотеки содержит описание работы с ней и необходимые ссылки



Библиотека **MapNavigator** предназначена для работы с картами **Google Maps**. Позволяет определять направления и отображать маршрут на карте. Работает только с **Google Maps v2**. Скачать можно на официальном сайте.





***AChartEngine*** - библиотека, предназначенная для построения графиков.

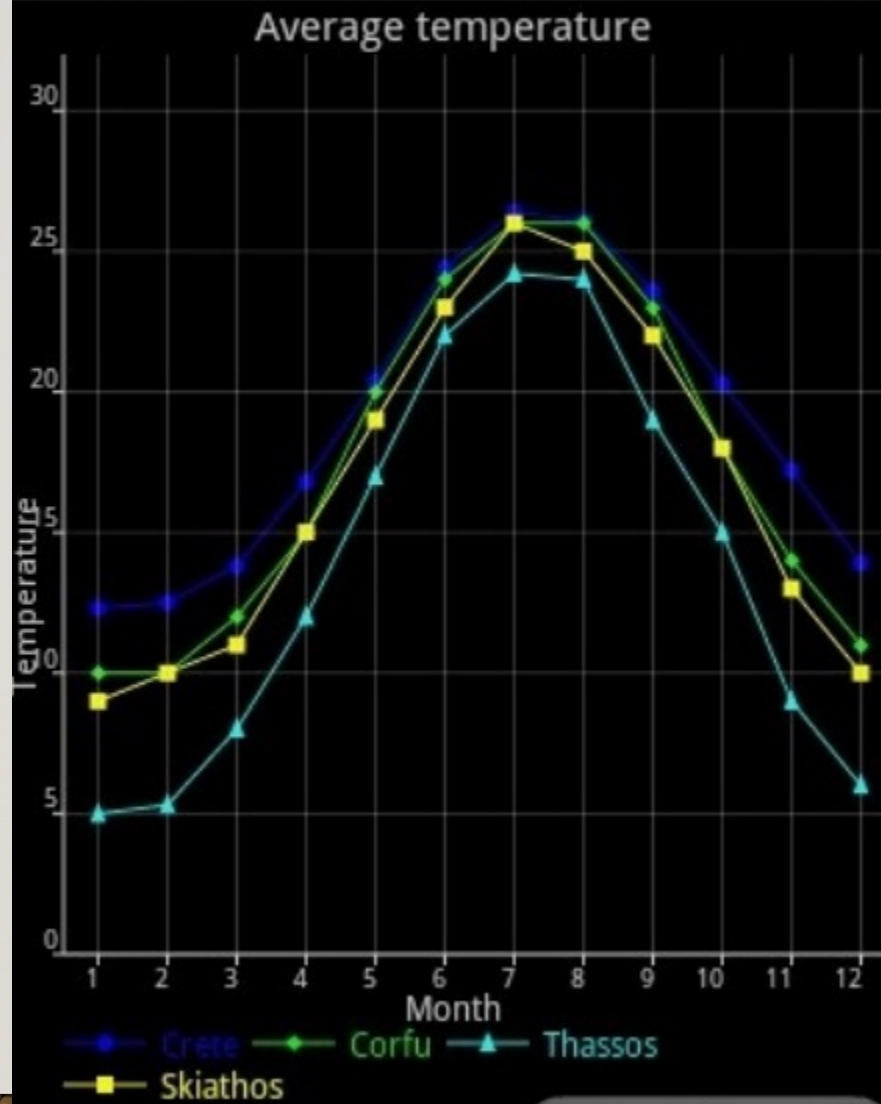
Позволяет строить графики различных типов:

- Линии графиков функций.
- Поточечные графики.
- Гистограммы.
- Круговые диаграммы.
- Пузырьковые диаграммы.
- Комбинированные диаграммы.
- Другие виды диаграмм и графиков.

Все типы диаграмм поддерживают несколько рядов данных. Сайт разработчика содержит подробную документацию, оформленную в стиле Javadoc pages, примеры использования библиотеки, а также ее исходный код.

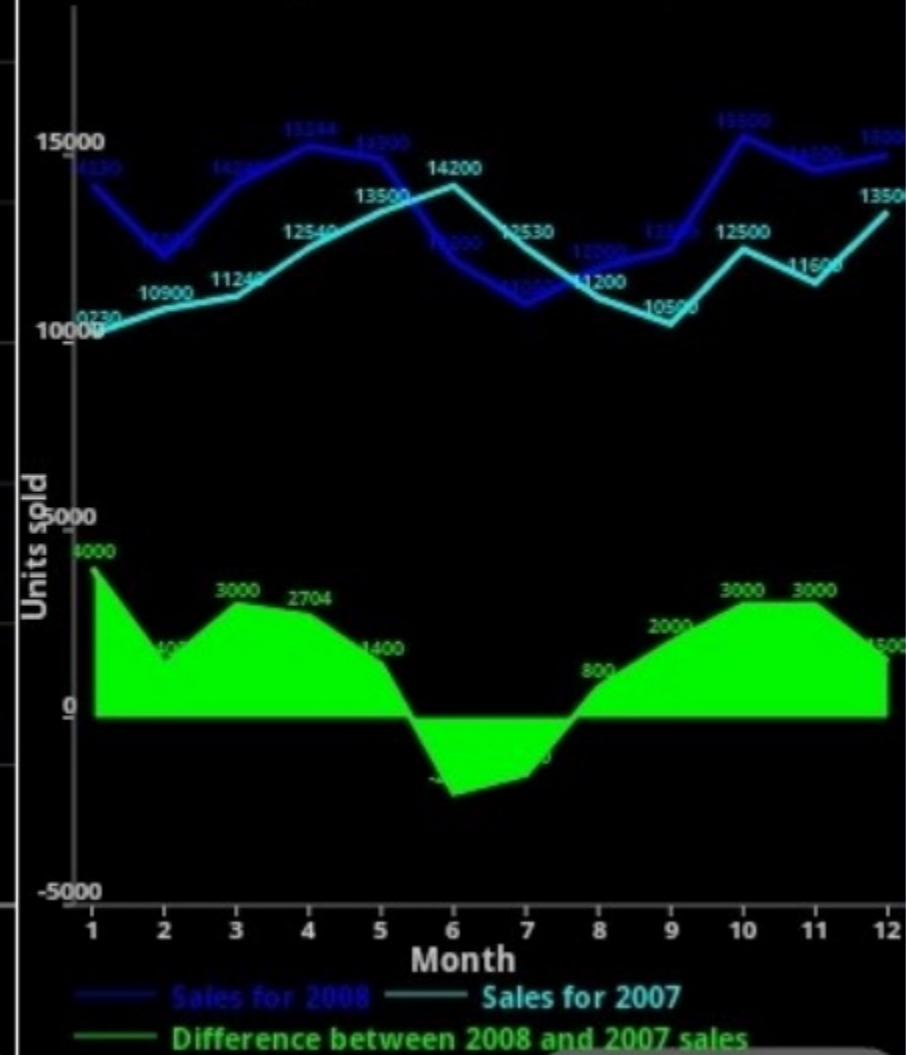


# Average temperature



# AChartEngine

## Monthly sales in the last 2 years



# Безопасность использования подключаемых библиотек

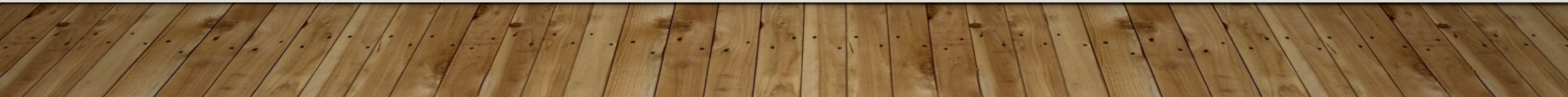
Подключаемые библиотеки являются очень удобным инструментом, облегчающим труд программиста. Однако разработчики приложений, использующие сторонние библиотеки подобного рода, часто не подозревают об их проблемах с безопасностью. Библиотека может содержать возможности, которые могут использоваться злоумышленниками в преступных целях.

При выборе библиотеки следует соблюдать осторожность. Поэтому при выборе библиотеки следует соблюдать ряд правил:





- Не использовать скомпрометированные библиотеки. Если о какойто библиотеке появляются сведения, что она может содержать вредоносный код, следует отказаться от ее использования в новых проектах и по возможности пересмотреть ее применение в уже существующих.
- С осторожностью использовать библиотеки из сомнительных источников.
- Обязательно ознакомиться с форумami и сайтами, где могут обсуждаться библиотеки. Кроме того, это может помочь вам подобрать наиболее подходящее решение для вашей конкретной задачи.
- По возможности просмотреть исходники.
- Применять другие правила информационной безопасности, которые могут иметь значение в каждом конкретном случае.



# База данных и мультимедиа в **Android**

Современное программирование трудно представить без использования баз данных, рано или поздно в процессе развития приложения появляется осознание необходимости долговременного хранения и обработки структурированной информации.

Рассмотрим вопросы, связанные с использованием баз данных **SQLite** в приложениях, разрабатываемых под **Android**. Базы данных **SQLite** являются основой построения рабочей и функциональной программы, в которой необходимо работать с большими объемами структурированной информации.



# Основы работы с базами данных, SQLite

SQLite - небольшая и при этом мощная система управления базами данных. Эта система создана в 2000 году, ее разработчик доктор Ричард Хипп (***Dr. Richard Hipp***). В настоящее время является одной из самых распространенных ***SQL-систем*** управления базами данных в мире. Можно выделить несколько причин такой популярности ***SQLite***: она бесплатная; она маленькая, примерно 150 Кбайт; не требует установки и администрирования. Подробнее см. . База данных ***SQLite*** - это обычный файл, его можно перемещать и копировать на другую систему (например, с телефона на рабочий компьютер) и она будет отлично работать. Android хранит файл базы данных приложения в папке :  
***data/data/packageName/databases/***,  
где ***package name*** - имя пакета, в котором расположено приложение.



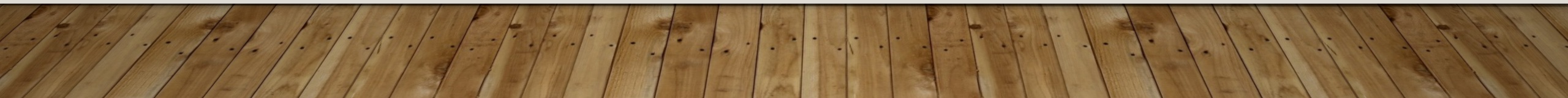
Обращения к базе данных **SQL** выполняются посредством запросов, существует три основных вида **SQL** запросов: **DDL**, **Modification** и **Query**. • **DDL** запросы. Такие запросы используются для создания таблиц. Каждая таблица характеризуется именем и описанием столбцов, которое содержит имя столбца и тип данных. В файле базы данных может быть несколько таблиц.

Пример запроса для создания таблицы:

```
create Table_Name ( _id integer primary key autoincrement,  
    field_name_1 text, field_name_2 text);
```



Первый столбец обозначен, как **primary key** (первичный ключ), т.е. уникальное число, которое однозначно идентифицирует строку. Слово **autoincrement** указывает, что база данных будет автоматически увеличивать значение ключа при добавлении каждой записи, что и обеспечивает его уникальность. Существует договоренность первый столбец всегда называть **\_id**, это не жесткое требование **SQLite**, однако может понадобиться при использовании контент-провайдера в **Android**. Стоит иметь в виду, что в **SQLite**, в отличие от многих других баз данных, типы данных столбцов являются лишь подсказкой, т. е. не вызовет никаких нареканий попытка записать строку в столбец, предназначенный для хранения целых чисел или наоборот. Этот факт можно рассматривать, как особенность базы данных, а не как ошибку, на это обращают внимание авторы **SQLite**.



- **Modification запросы.** Такие запросы используются для добавления, изменения или удаления записей. Пример запроса на добавление строки:

***insert into Table\_Name values(null, value1, value2);***

В этом случае значения разместятся в соответствующие столбцы таблицы, первое значение задается для поля ***\_id*** и равно ***null***, т. к. ***SQLite*** вычисляет значение этого поля самостоятельно.





При добавлении можно указывать столбцы, в которые будут размещаться значения, остальные столбцы заполнятся значениями по умолчанию, в этом случае можно добавлять элементы в измененном порядке. Пример такого запроса:

```
insert into Table_Name(field_name_2, field_name_1) values(value2, value1);
```

В этом случае добавляются значения только в поля ***field\_name\_1*** и ***field\_name\_2***, причем изменен порядок следования полей, а вместе с этим и порядок следования значений, иногда это бывает удобно.



## Примеры запросов на изменение строки:

***update Table\_Name set Field\_Name\_I = value;***

поменяет значение столбца ***Field\_Name\_I*** на ***value*** во всей таблице;

***update Table\_Name set Field\_Name\_I = value where \_id = smth;***

поменяет значение столбца ***Field\_Name\_I*** только в той строке, ***\_id*** которой равен ***smth***.

## Примеры запросов на удаление строк:

***delete from Table\_Name;***

***delete from Table\_Name where Field\_Name\_I = smth;***

первый запрос удаляет таблицу целиком, второй - только те строки, в которых столбец ***Field\_Name\_I*** имеет значение ***smth***.

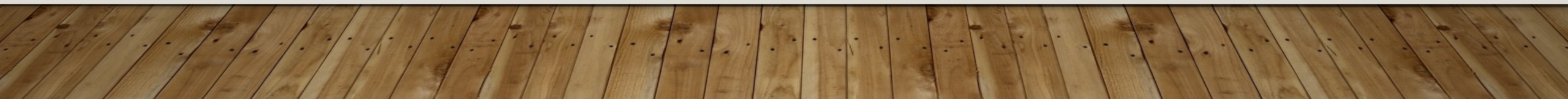
## Query запросы.

Такие запросы позволяют получать выборки из таблицы по различным критериям.

Пример запроса:

```
select * from Table_Name where (_id = smth);  
select Field_Name_1, Field_Name_2 from Table_Name Field_Name_1 =  
smth);
```

Первый запрос выводит строку с ***\_id*** равным ***smth***, второй - выводит два элемента ***Field\_Name\_1*** и ***Field\_Name\_2*** строк, в которых ***Field\_Name\_1*** равен ***smth***.





Любая база данных, созданная в приложении доступна любому классу приложения, но недоступна из вне. Чтобы открыть доступ к базе данных другим приложениям необходимо использовать контент-провайдеры (**Content Providers**). Для создания и обновления базы данных в **Android** предусмотрен класс **SQLiteOpenHelper**. При разработке приложения, работающего с базами данных, необходимо создать класс-наследник от **SQLiteOpenHelper**, в котором обязательно реализовать методы:

**onCreate()** - вызывается при первом создании базы данных; **onUpgrade()** - вызывается, когда необходимо обновить базу данных. По желанию можно реализовать метод: **onOpen()** - вызывается при открытии базы данных.



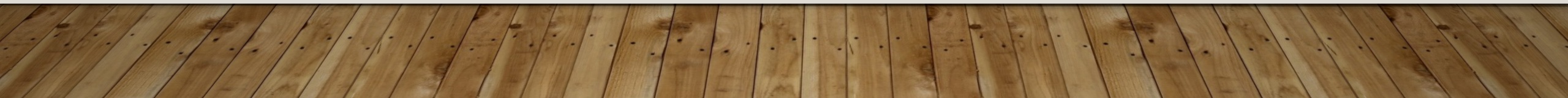
В **Android** предусмотрен класс для работы с базой данных **SQLite** напрямую, этот класс называется **SQLiteDatabase** и содержит методы:  
**openDatabase()** - позволяет открыть базу данных;

- update()** - позволяет обновить строки таблицы базы данных;
- insert()** - позволяет добавлять строки в таблицу базы данных;
- delete()** - позволяет удалять строки из таблицы базы данных;
- query()** - позволяет составлять запросы к базе данных;
- execSQL()** - позволяет выполнять запросы к базе данных.



Для добавления новых строк в таблицу используется класс ***ContentValues***, каждый объект этого класса представляет собой одну строку таблицы и выглядит как ассоциативный массив с именами столбцов и значениями, которые им соответствуют.

Для получения результатов запросов к базе данных используется класс ***Cursor***, объекты этого класса ссылаются на результирующий набор данных, позволяют управлять текущей позицией в возвращаемом при запросе наборе данных.





Для предоставления доступа к данным для других приложений можно использовать контент-провайдеры (**ContentProvider**). Любая информация, управляемая контент-провайдером адресуется посредством **URI**:

**content://authority/path/id** где: **content://** - стандартный требуемый префикс;

**authority** - имя провайдера, рекомендуется использовать полное квалификационное имя пакета для избежания конфликта имен;

**path** - виртуальная папка внутри провайдера, которая определяет вид запрашиваемых данных;

**id** - первичный ключ отдельной запрошенной записи, для запроса всех записей определенного типа этот параметр не указывается

