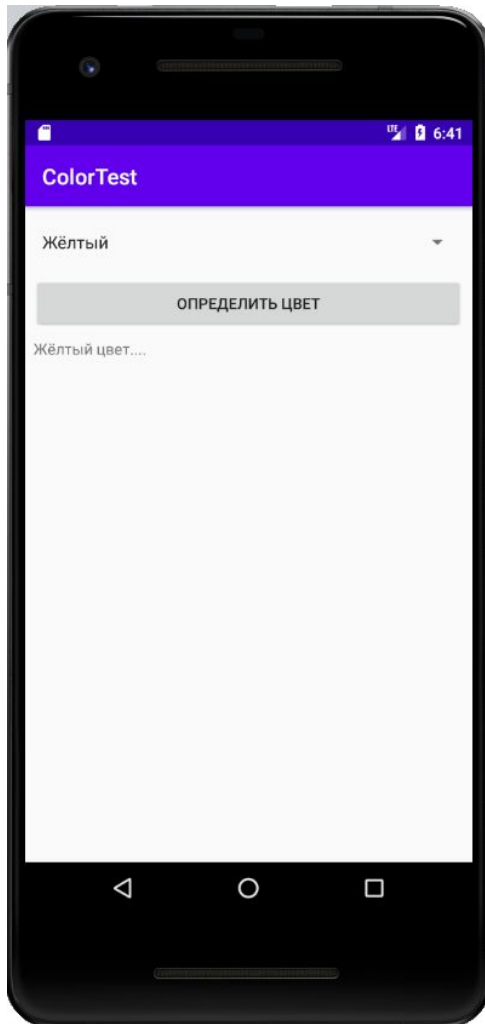


**Лекция №5**  
**по курсу «ImageView, radioButton, ИНТЕНТЫ»**

Москва 2021

# Программа Выбора цветов



```
android:paddingLeft="16dp"  
android:paddingRight="16dp"  
android:paddingTop="16dp"  
android:paddingBottom="16dp"
```

# Определение размеров

Android – очень фрагментированная система. Это означает, что устройства, работающие на ОС Android имеют различные размеры, разрешения и соотношения сторон, форм-факторы, железную начинку, сенсоры, вендорский софт и версии ОС.

**Размер экрана** – это физический размер экрана, измеряется диагональ. Для удобства, Android группирует все существующие размеры экранов в 4 общие группы:

- маленькие – small,
- обычные – normal,
- большие – large,
- очень большие extra-large.

**Плотность экрана** – количество пикселей на определенной площади экрана, обычно определяется как количество точек на дюйм (dpi – dots per inch).

Для удобства, Android группирует все существующие плотности экрана в 6 общих групп:

- ldpi (low) ~120dpi
- mdpi (medium) ~160dpi
- hdpi (high) ~240dpi
- xhdpi (extra-high) ~320dpi
- xxhdpi (extra-extra-high) ~480dpi
- xxxhdpi (extra-extra-extra-high) ~640dpi

# Определение размеров

Для того чтобы интерфейс был более-менее идентичен на различных экранах, были придуманы так называемые пиксели, не зависящие от плотности, или dp (дипы).

dp рассчитываются по формуле:

**dp = длина в пикселях \* 160 / плотность экрана**

Следовательно, на экране с плотностью 160 точек на дюйм, 1 dp = 1 px.

И наоборот, чтобы из dp получить пиксели, нужно воспользоваться формулой

$px = dp * \text{плотность экрана} / 160$

Так, на экране с плотностью 240 dpi, 100 dp будут занимать 150 px.

px – пиксели, в них чаще всего указывается разрешение экрана (например. 1920x1080). Элементы интерфейса нежелательно указывать в пикселях, но разумеется, они будут выглядеть по-разному, могут иметь различный размер на разных экранах. С другой стороны, именно в пикселях возвращают значения методы `getWidth()` и `getHeight()`, поэтому полностью избежать работы с пикселями проблематично.

# Определение размеров

Рекомендуется использовать именно dp, чтобы ваша верстка выглядела более менее одинаково на всех экранах.

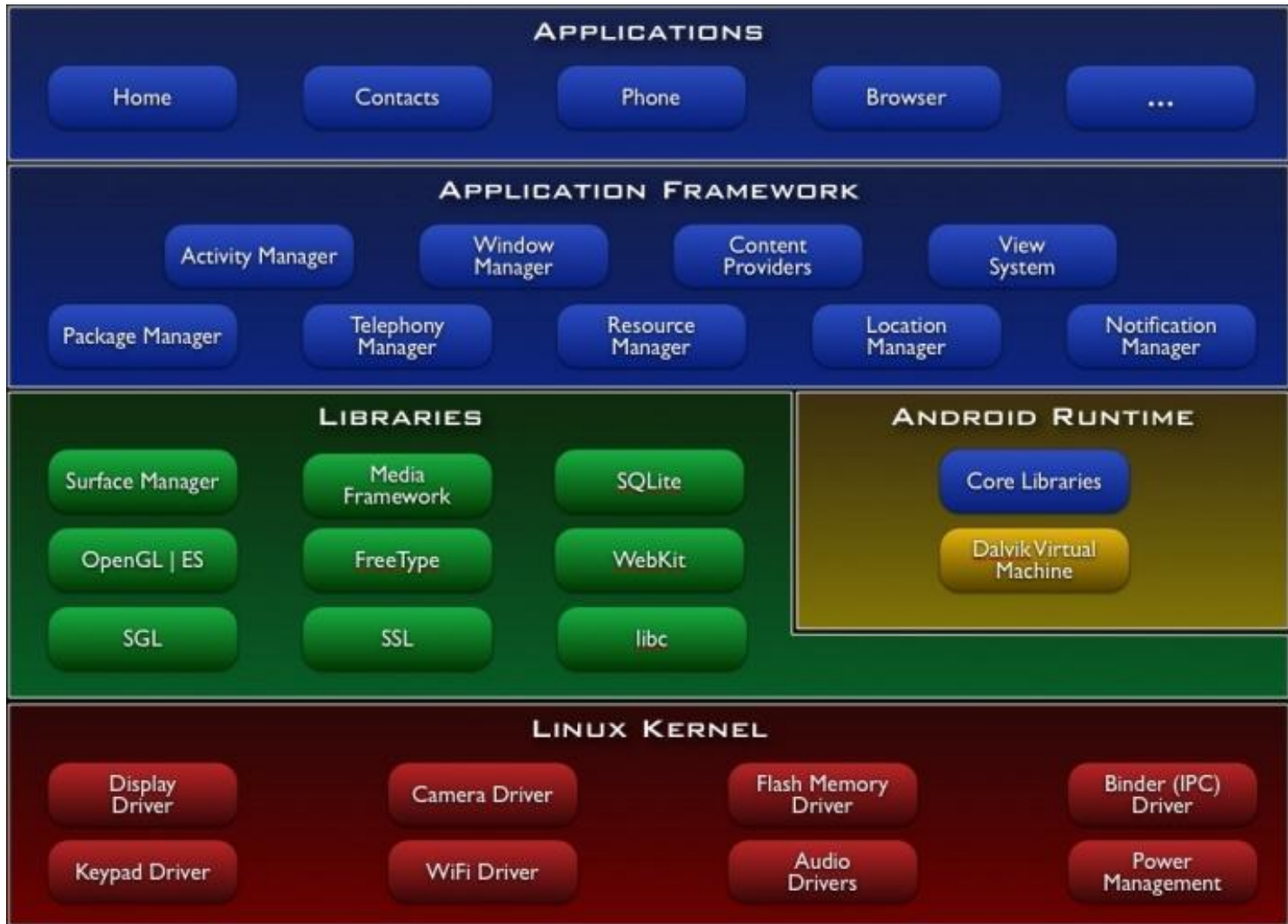
Также нужно сказать и про другую единицу измерения, sp.

sp – это пиксели, не зависящие от масштаба, главным образом применяются при указании размеров шрифтов текста.

## Документация

[https://developer.android.com/guide/practices/screens\\_support.html](https://developer.android.com/guide/practices/screens_support.html)

# Архитектура Android



# Архитектура Android

Уровень абстракции между аппаратным обеспечением и программным стеком:

- В основе лежит ядро ОС Linux (несколько урезанное)
- Обеспечивает функционирование системы;
- Отвечает за безопасность;
- Управляет памятью, энергосистемой и процессами;
- Предоставляет сетевой стек и модель драйверов.



## Набор библиотек

Обеспечивает важнейший базовый функционал для приложений:

- Алгоритмы для вышележащих уровней
- Поддержка файловых форматов
- Кодирование и декодирование информации
- Отрисовка графики и т. д.





# Библиотеки

Реализованы на C/C++ и скомпилированы под конкретное аппаратное обеспечение устройства, вместе с которым они поставляются:

- **Surface Manager**
- **Media Framework**
- **SQLite**
- **3D библиотеки**
- **LibWebCore**
- **SGL (Skia Graphics Engine)**
- **SSL**

# Среда выполнения Android Runtime

- Библиотеки ядра, обеспечивающие большую часть низкоуровневой функциональности, доступной библиотекам ядра языка Java
- Виртуальная машина Dalvik, позволяющая запускать приложения



## Application Framework – уровень каркаса приложений

- Обеспечивает разработчикам доступ к API, предоставляемым компонентами системы уровня библиотек
- Любому приложению предоставляются уже реализованные возможности других приложений, к которым разрешено получать доступ

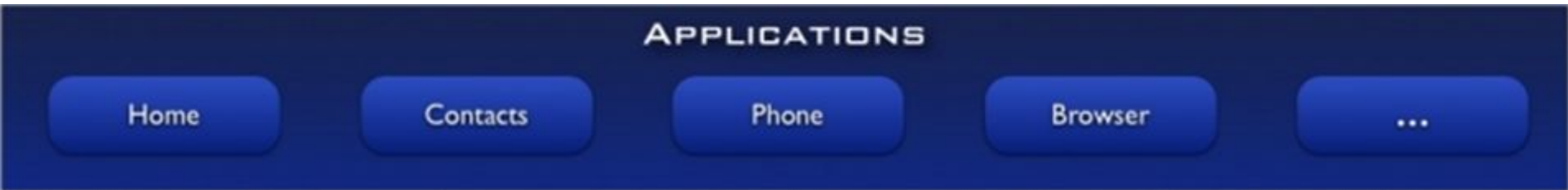


## Application Framework

- Богатый и расширяемый набор представлений (**Views**)
- Контент-провайдеры (**Content Providers**)
- Менеджер ресурсов (**Resource Manager**)
- Менеджер оповещений (**Notification Manager**)
- Менеджер действий (**Activity Manager**)
- Менеджер местоположения (**Location Manager**)

## Applications – уровень приложений

- Набор базовых приложений, который предустановлен на ОС Android. Например, браузер, почтовый клиент, программа для отправки SMS, карты, календарь, менеджер контактов и др.



## Виды приложений

1. Приложения переднего плана
2. Фоновые приложения
3. Смешанные приложения
4. Виджеты

Приложения под Android разрабатываются на языке программирования Java, компилируется в файл с расширением .apk, после этот файл используется для установки приложения на устройства, работающие под управлением Android.

## Атрибут weightSum

**android:weightSum – какую часть в панели контейнере занимает каждый из находящихся в ней компонентов**

**weightSum полезен для правильного отображения макета для любого устройства, чего не произойдет, если вы установите ширину и высоту напрямую.**

Согласно документации, android:weightSum определяет максимальную сумму веса и рассчитывается как сумма layout\_weight всех дочерних элементов, если это не указано явно.

Рассмотрим пример с LinearLayout горизонтальной ориентацией и 3 ImageViews внутри него. Теперь мы хотим, чтобы они ImageViews всегда занимали одинаковое пространство. Чтобы добиться этого, вы можете установить layout\_weight для каждого ImageView значение 1, и weightSum будет вычислено значение 3

Если в LinearLayout есть 2 представления, первое с layout\_weight, равным 1, второе с layout\_weight, равным 2 и не указанное weightSum, по умолчанию, weightSum рассчитывается как 3 (сумма весов дочерних элементов) и первый вид занимает  $1/3$  пространства, а второй -  $2/3$ .

Однако, если бы мы указали weightSum как 5, первое заняло бы  $1/5$  пространства, а второе -  $2/5$ . Таким образом, в общей сложности  $3/5$  пространства будет занято макетом, а остальное останется пустым.

# ImageView

Для ImageView можно задать правило для масштабирования в разметке через атрибут `scaleType` или программно через метод `setScaleType()`

## **Варианты значений:**

- `fitXY`
- `fitCenter`
- `fitStart`
- `fitEnd`
- `center`
- `centerCrop`
- `centerInside`
- `matrix`



# Приложение страны

