Макеты и Активности

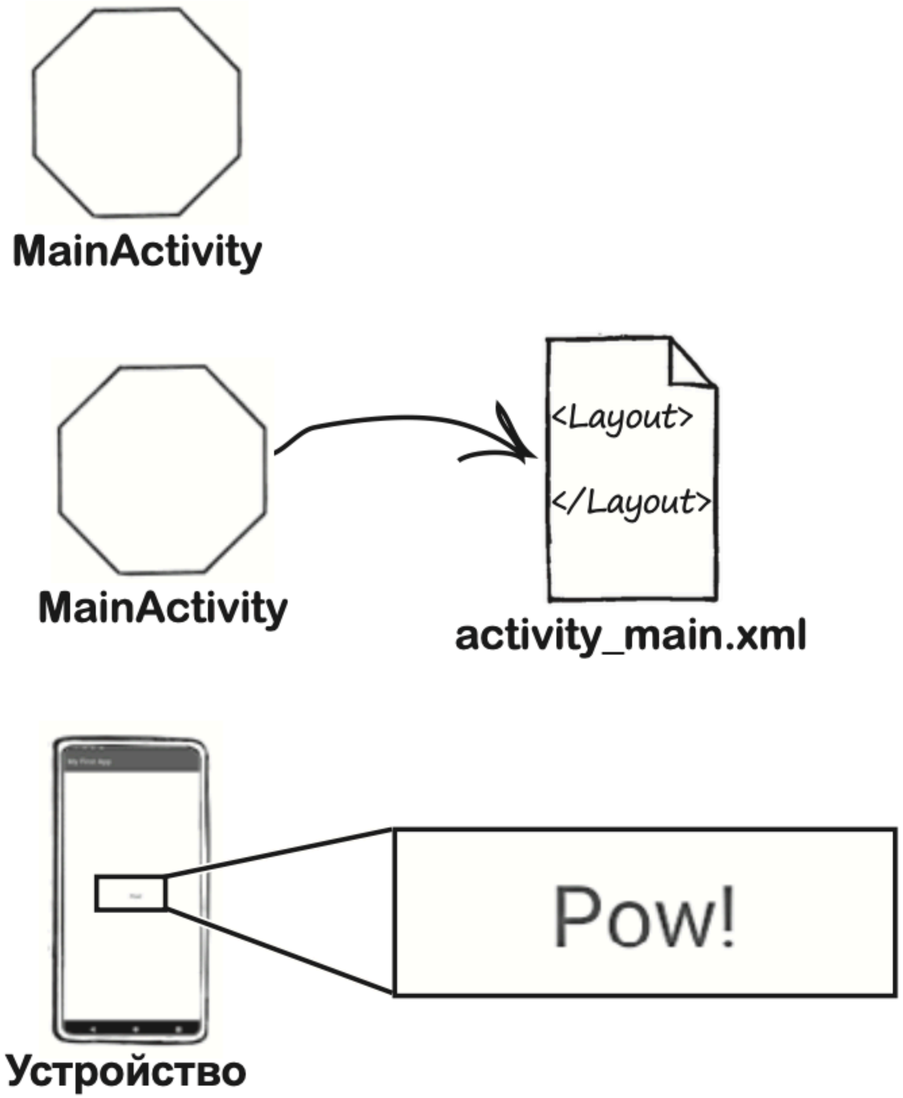
Layout and Activity

S. Chebotarev 2023

Экран приложения

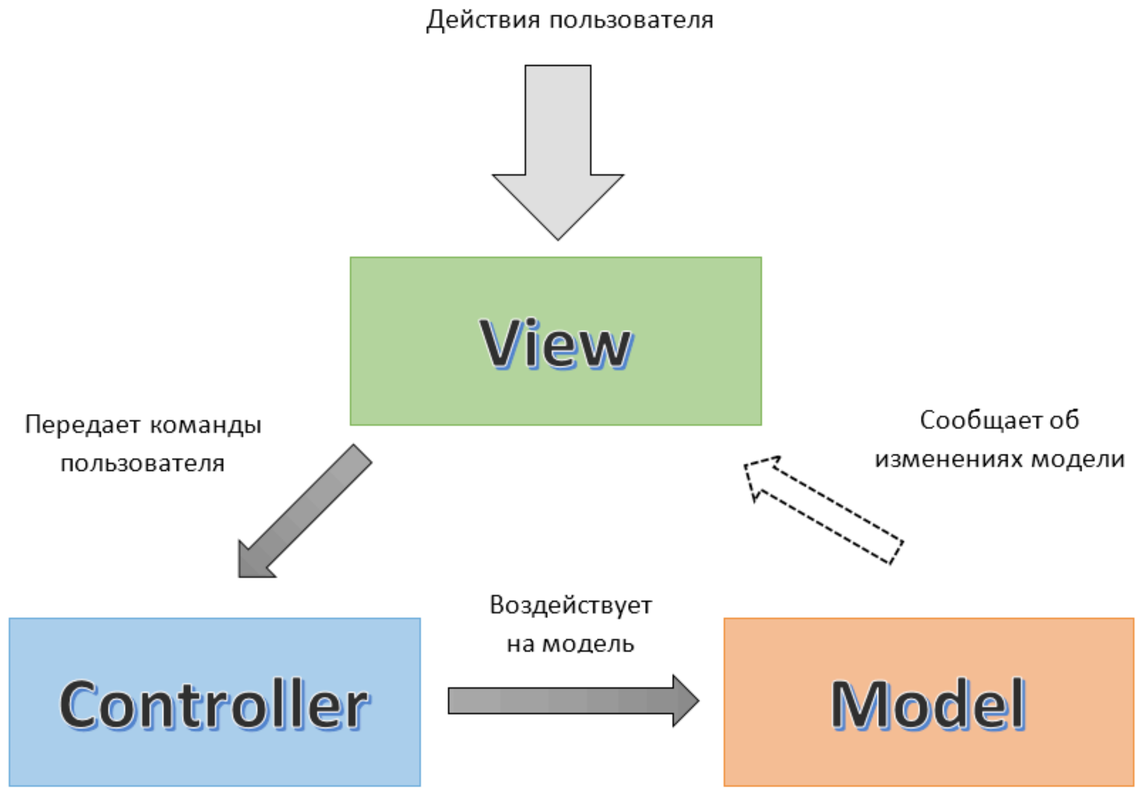
Activity + Layout

- Activity - Kotlin класс
- Layout - XML

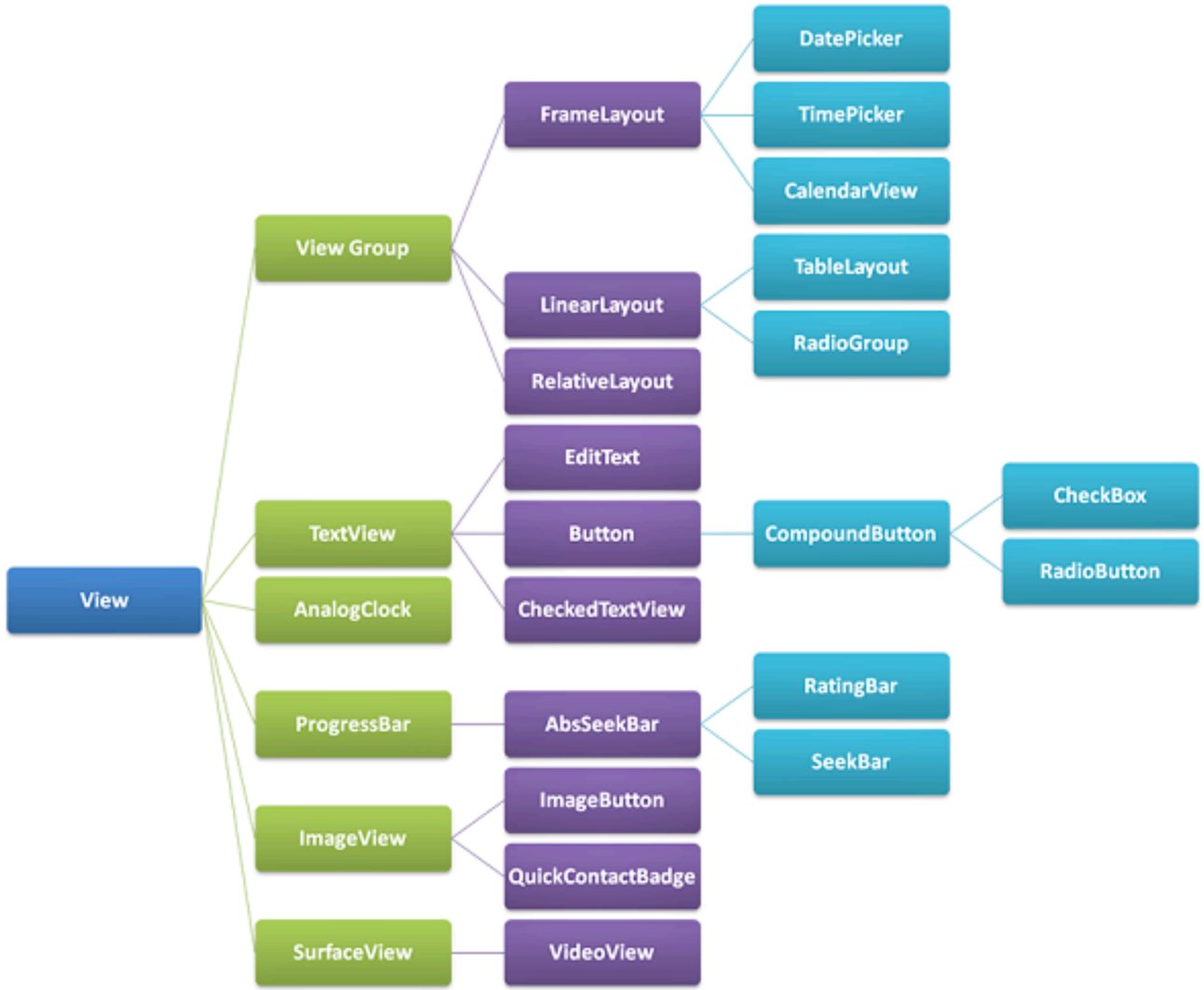


MVC

- View
- Controller
- ~~Model~~



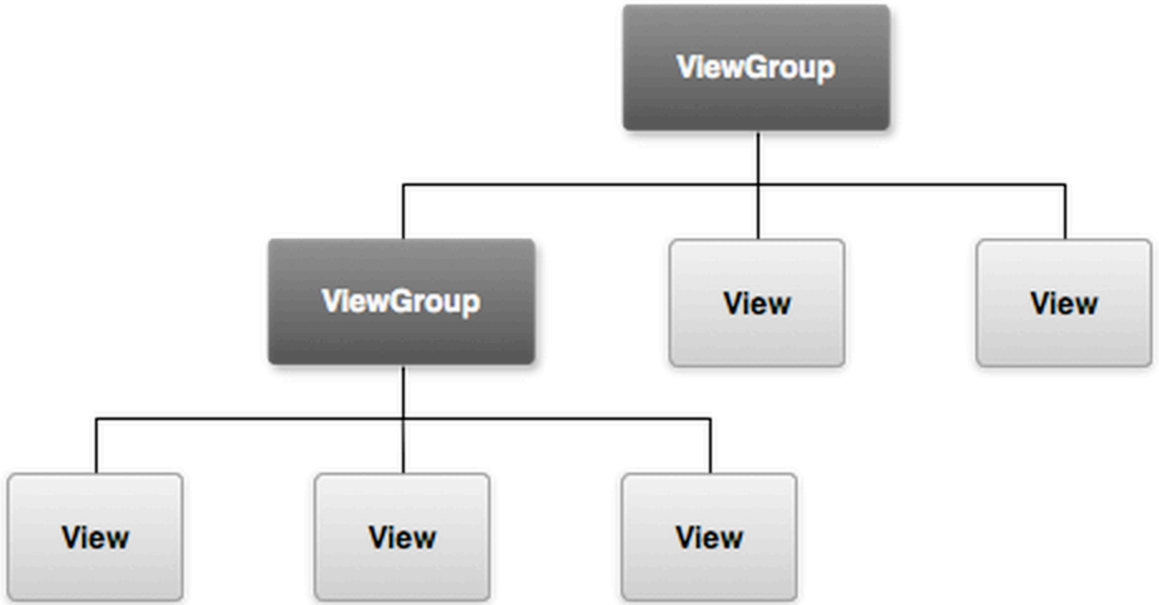
Иерархия View



View

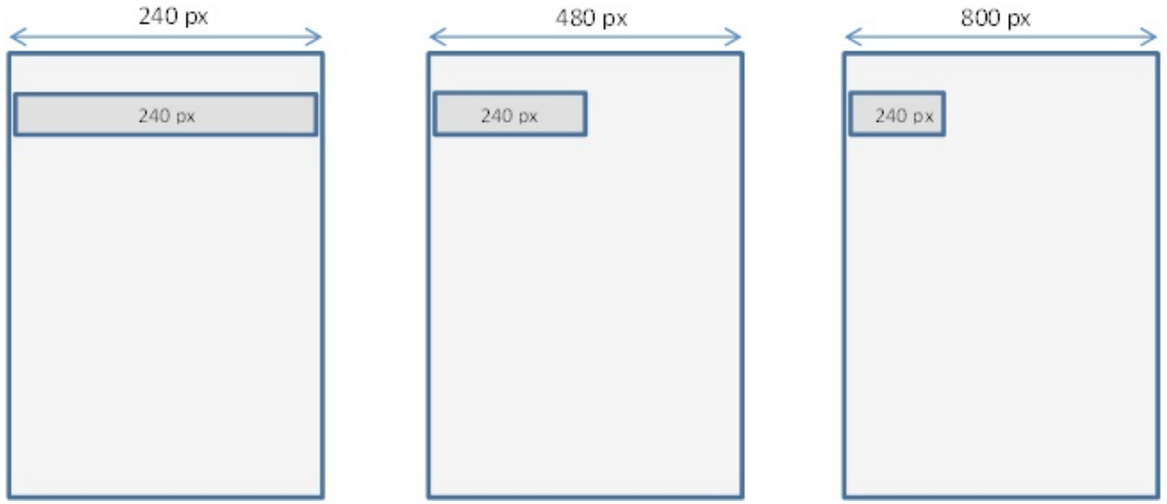
Компоненты графического интерфейса

- View
- GroupView

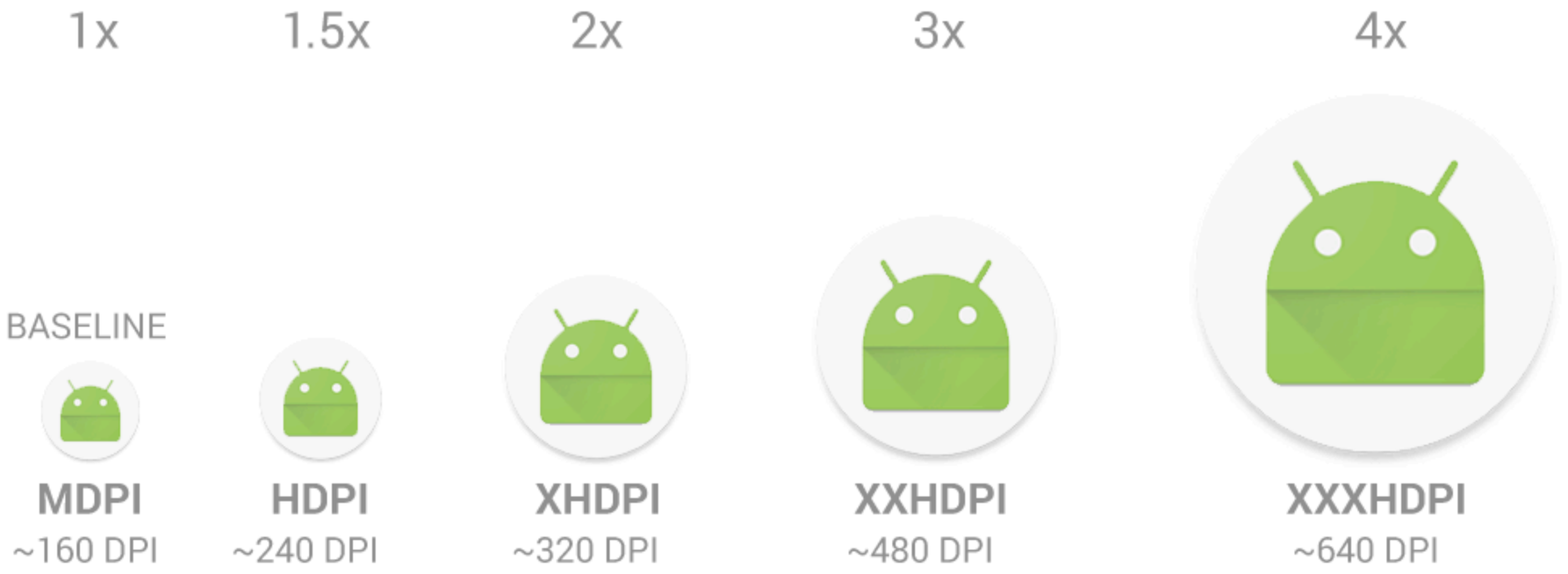


DP и SP

- DP - Density-independent Pixels
- SP - Scale-independent Pixels
-



`ldpi`	Resources for low-density (*ldpi*) screens (~120 dpi).
`mdpi`	Resources for medium-density (*mdpi*) screens (~160 dpi). This is the baseline density.
`hdpi`	Resources for high-density (*hdpi*) screens (~240 dpi).
`xhdpi`	Resources for extra-high-density (*xhdpi*) screens (~320 dpi).
`xxhdpi`	Resources for extra-extra-high-density (*xxhdpi*) screens (~480 dpi).
`xxxhdpi`	Resources for extra-extra-extra-high-density (*xxxhdpi*) uses (~640 dpi).
`nodpi`	Resources for all densities. These are density-independent resources. The system doesn't scale resources tagged with this qualifier, regardless of the current screen's density.
`tvdpi`	Resources for screens somewhere between mdpi and hdpi; approximately ~213 dpi. This isn't considered a "primary" density group. It is mostly intended for televisions, and most apps don't need it—providing mdpi and hdpi resources is sufficient for most apps, and the system scales them as


1x
1.5x
2x
3x
4x
BASELINE
MDPI
~160 DPI
HDPI
~240 DPI
XHDPI
~320 DPI
XXHDPI
~480 DPI
XXXHDPI
~640 DPI

```kotlin
//px = dp * (dpi / 160)
val sizeDP = 100.0f
val sizePX = TypedValue.applyDimension(
    TypedValue.COMPLEX_UNIT_DIP,
    value: sizeDP + 0.5F,
    resources.displayMetrics
).toInt()
val sizeDPNew =  sizePX / resources.displayMetrics.scaledDensity
```