**Стили**

Стили и темы в Android позволяют отделить детали дизайна вашего приложения от структуры и поведения пользовательского интерфейса.

Стиль — это набор атрибутов, определяющий внешний вид элемента пользовательского интерфейса . Стиль может указывать такие атрибуты, как цвет шрифта, размер шрифта, цвет фона и многое другое.

Тема — это набор атрибутов, которые применяются ко всему приложению или экрану, а не только к отдельному элементу. Когда вы применяете тему, каждый элемент в приложении или на экране применяет все поддерживаемые атрибуты темы. Темы также могут применять стили к таким элементам, как например, к строке состояния или фону окна.

Стили и темы объявляются в файле ресурсов стиля в res/values/ , обычно называемом styles.xml .

Две темы примененные к одному и тому же экрану

|  |  |
| --- | --- |

Темы и стили имеют одинаковую базовую структуру — пару ключ-значение, которая сопоставляет атрибуты с ресурсами .

Стиль определяет атрибуты для определенного типа элемента. Например, один стиль может определять атрибуты кнопки. Каждый атрибут, указанный вами в стиле, является атрибутом, который вы можете установить в файле разметки. Извлечение всех атрибутов стиля упрощает их использование и поддержку в нескольких виджетах.

Тема определяет коллекцию именованных ресурсов, на которые могут ссылаться стили, макеты, виджеты и т. д. Темы присваивают ресурсам Android семантические имена, например colorPrimary .

Стили и темы предназначены для совместной работы. Например, у вас может быть стиль, указывающий, что одна часть кнопки — colorPrimary , а другая — colorSecondary . Фактические определения этих цветов представлены в теме. Когда устройство переходит в ночной режим, ваше приложение может переключиться со «светлой» темы на «темную», изменяя значения для всех этих имен ресурсов. Вам не нужно менять стили, поскольку в стилях используются семантические имена, а не конкретные определения цветов.

**Создание и применение стиля**

Чтобы создать новый стиль, откройте файл res/values/styles.xml вашего проекта. Для каждого стиля, который вы хотите создать, выполните следующие действия:

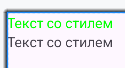
* Добавьте элемент <style> с именем, которое однозначно идентифицирует стиль.
* Добавьте элемент <item> для каждого атрибута стиля, который вы хотите определить. name в каждом элементе указывает атрибут, который вы в противном случае используете в качестве атрибута XML в макете. Значение элемента <item> является значением этого атрибута.

Например, предположим, что вы определяете следующий стиль:



Вы можете применить стиль к виду следующим образом:

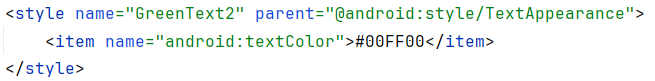


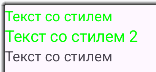


Каждый атрибут, указанный в стиле, применяется к этому представлению, если представление его принимает. Представление игнорирует любые атрибуты, которые оно не принимает.

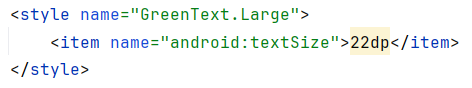
При создании собственных стилей всегда расширяйте существующий стиль из платформы или библиотеки Android, чтобы обеспечить совместимость со стилями пользовательского интерфейса платформы. Чтобы расширить стиль, укажите стиль, который вы хотите расширить, с помощью parent атрибута. Затем вы можете переопределить унаследованные атрибуты стиля и добавить новые.

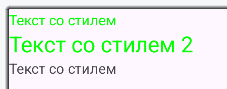
Например, вы можете унаследовать внешний вид текста платформы Android по умолчанию и изменить его следующим образом:





Вы также можете наследовать стили, расширив имя стиля точечной нотацией вместо использования parent атрибута. Добавьте к имени вашего стиля имя стиля, который вы хотите унаследовать, через точку. Например, следующий стиль наследует все стили GreenText из предыдущего примера, а затем увеличивает размер текста:





Вы можете продолжать наследовать подобные стили столько раз, сколько захотите, добавляя больше имен. Если вы используете точечную нотацию для расширения стиля и также включаете parent атрибут, то родительские стили переопределяют любые стили, унаследованные через точечную нотацию.

**Темы**

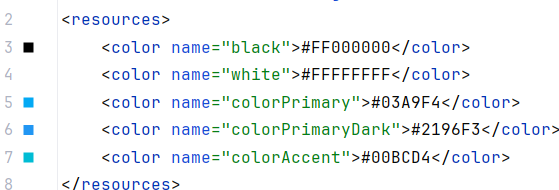
Вы можете создать тему так же, как и стиль. Разница в том, как вы его применяете: вместо применения стиля с атрибутом style к элементу вы применяете тему с атрибутом android:theme либо к тегу <application> , либо к тегу <activity> в файле AndroidManifest.xml.

Например, вот как применить «темную» тему Material Design из библиотеки Android ко всему приложению, но применить светлую к определенному действию:



Например создадим собственную тему для приложения которую будут использовать все элементы

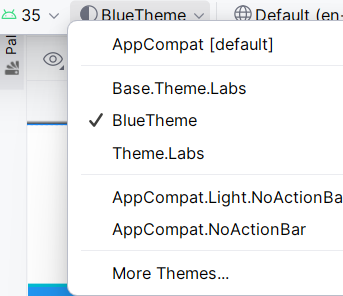
Создайте дополнительные цвета в файле colors.xml цвета для темы на ваше усмотрение



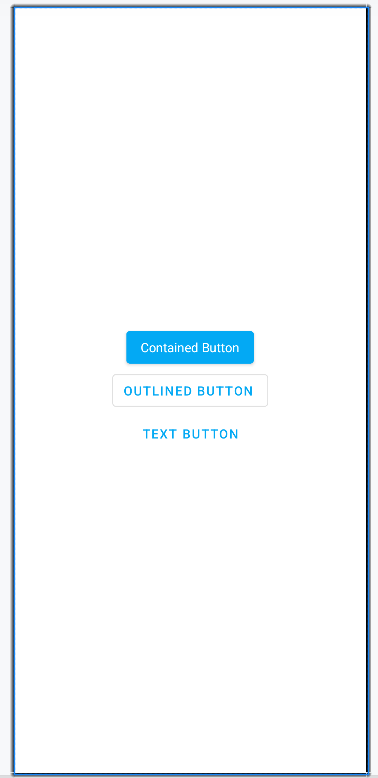
Создайте в файле themes.xml новую тему, она использует созданные ранее цвета, пусть тема наследует базовую светлую тему



Установите созданную тему и создайте несколько кнопок на вашем макете экрана







Также тему можно изменить программно с помощью метода **setTheme()** который принимает значение ресурса темы.

**Иерархия стилей**

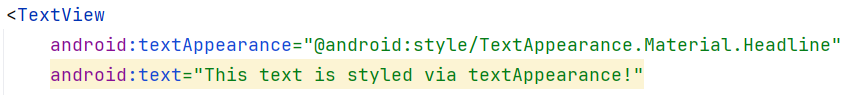
Android предоставляет множество способов установки атрибутов в вашем приложении. Например, вы можете установить атрибуты непосредственно в макете, применить стиль к элементу, применить тему к макету или установить атрибуты программным способом.

Если вы стилизуете свое приложение и не видите ожидаемых результатов, вполне вероятно, что другие стили переопределяют ваши изменения. Например, если вы применяете тему к своему приложению вместе со стилем к отдельному View , атрибуты стиля переопределяют любые соответствующие атрибуты темы для этого View . Однако обратите внимание, что любые атрибуты темы, которые не переопределяются стилем, по-прежнему используются.

Выбирая стиль своего приложения, помните об иерархии стилей Android. В общем, используйте темы и стили как можно чаще, чтобы обеспечить единообразие. Если вы укажете одни и те же атрибуты в нескольких местах, следующий список определит, какие атрибуты будут применены в конечном итоге. Список упорядочен от самого высокого приоритета к самому низкому:

* программная установка значений атрибутам элементов;
* применение атрибутов непосредственно к элементу.
* применение стиля к элементу;
* применение стиля к макету;
* применение стиля к приложению, коллекции элементов;
* применение определенного стиля, специфичного для элемента, например установка TextAppearance для TextView.

Одним из ограничений стилей является то, что к элементу можно применить только один стиль. Однако это неверно для некоторых атрибутов которые устанавливают стиль для отдельных компонентов элемента. Например, в TextView вы также можете указать атрибут TextAppearance, который действует аналогично стилю, как показано ниже:



TextAppearance позволяет определить стиль, специфичный для текста, оставляя стиль элемента доступным для других целей. Однако, если вы определяете какие-либо текстовые атрибуты непосредственно в элементе или в стиле, эти значения переопределяют значения TextAppearance.

**Задание для самостоятельной работы:**

1. Разработайте приложение с интерфейсом соответствующим заданному макету. Разработайте для приложения темную тему, которую возможно переключать в том числе нажатием кнопки. Разработайте альтернативную тему (светлую и темную варианты) с другой цветовой палитрой, которую возможно переключать нажатием кнопки.

|  |  |
| --- | --- |