**Использование элементов для ввода данных**

**EditText**

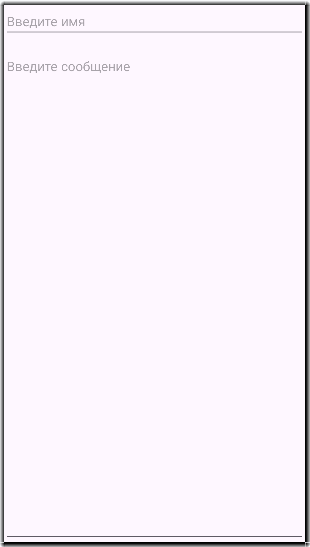
Элемент EditText является подклассом класса TextView. Он также представляет текстовое поле, но теперь уже с возможностью ввода и редактирования текста. Таким образом, в EditText мы можем использовать все те же возможности, что и в TextView.

Из тех атрибутов, что не рассматривались в теме про TextView, следует отметить атрибут android:hint. Он позволяет задать текст, который будет отображаться в качестве подсказки, если элемент EditText пуст. Кроме того, мы можем использовать атрибут android:inputType, который позволяет задать клавиатуру для ввода. В частности, среди его значений можно выделить следующие:

* text: обычная клавиатура для ввода однострочного текста
* textMultiLine: многострочное текстовое поле
* textEmailAddress: обычная клавиатура, на которой присутствует символ @, ориентирована на ввод email
* textUri: обычная клавиатура, на которой присутствует символ /, ориентирована на ввод интернет-адресов
* textPassword: клавиатура для ввода пароля
* textCapWords: при вводе первый введенный символ слова представляет заглавную букву, остальные - строчные



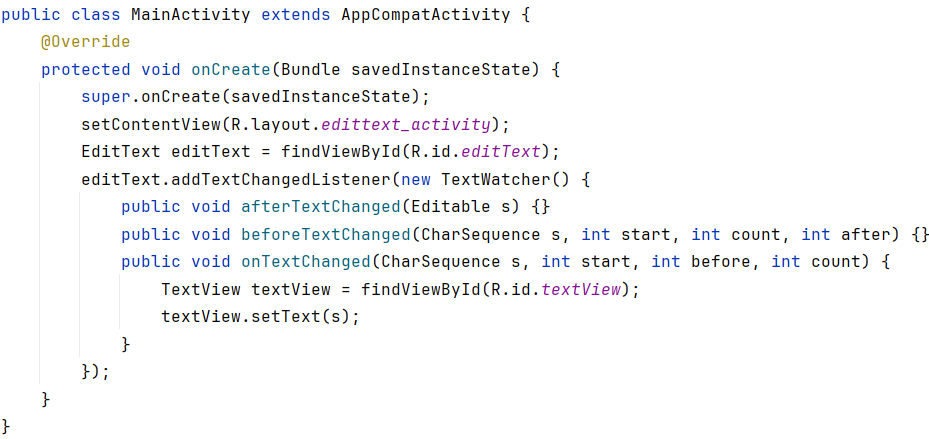
Первое поле обычное однострочное, а второе - многострочное. Чтобы во втором поле текст выравнивался по верху, дополнительно устанавливается атрибут android:gravity="top".



Одной из возможностей элемента EditText также является возможность обработать введенные символы по мере ввода пользователя. Для этого определим в файле activity\_main.xml следующую разметку:

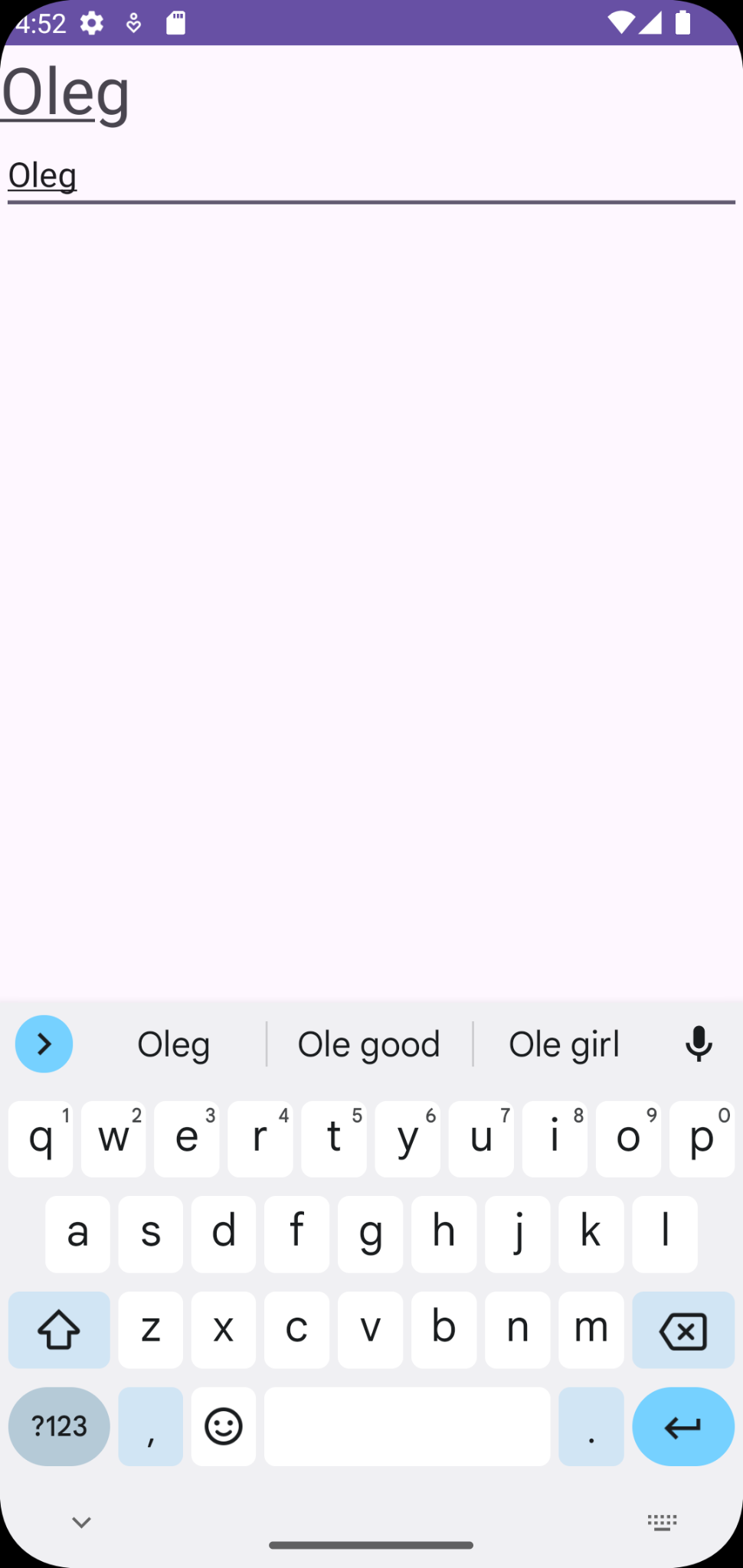


Предполагается, что введенные в EditText символы тут же будут отображаться в элементе TextView. И для этого также изменим код MainActivity:



С помощью метода addTextChangedListener() здесь к элементу EditText добавляется слушатель ввода текста - объект TextWatcher. Для его использования нам надо реализовать три метода, но в реальности нам хватит реализации метода onTextChanged, который вызывается при изменении текста. Введенный текст передается в этот метод в качестве параметра CharSequence. В самом методе просто передаем этот текст в элемент TextView.

В итоге при вводе в EditText все символы также будут отображаться в TextView:



Рассмотрим некоторые дополнительные атрибуты:

Для изменения расположения текста в EditText вы можете использовать атрибут gravity



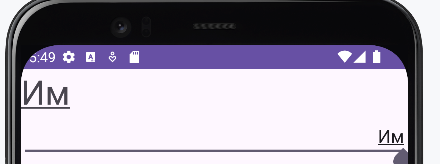


Для добавления текста в EditText вы можете использовать атрибут text





Но в случае попытки самостоятельно заполнить текст в EditText будет необходимо удалить и написать новый текст, что не всегда необходимо.

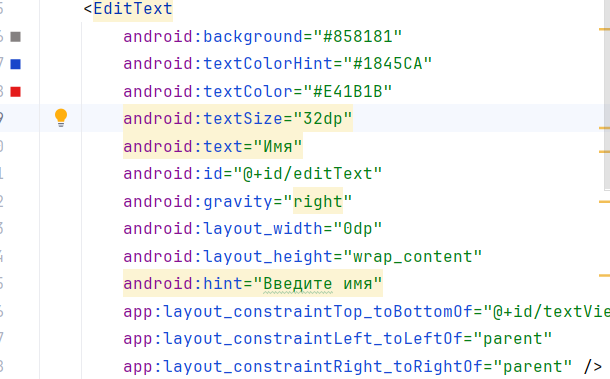


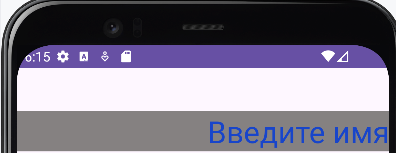
Для понимания пользователем какую именно информацию необходимо написать в EditText, используется атрибут hint который устанавливает текст полупрозрачной подсказки

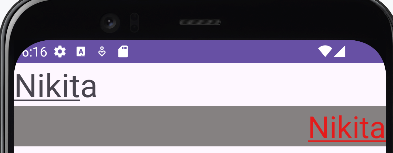




Вы можете менять большинство стандартный свойств элемента такие как цвет фона и стандартных свойств текста такие как цвет текста, но можете устанавливать и цвет подсказки



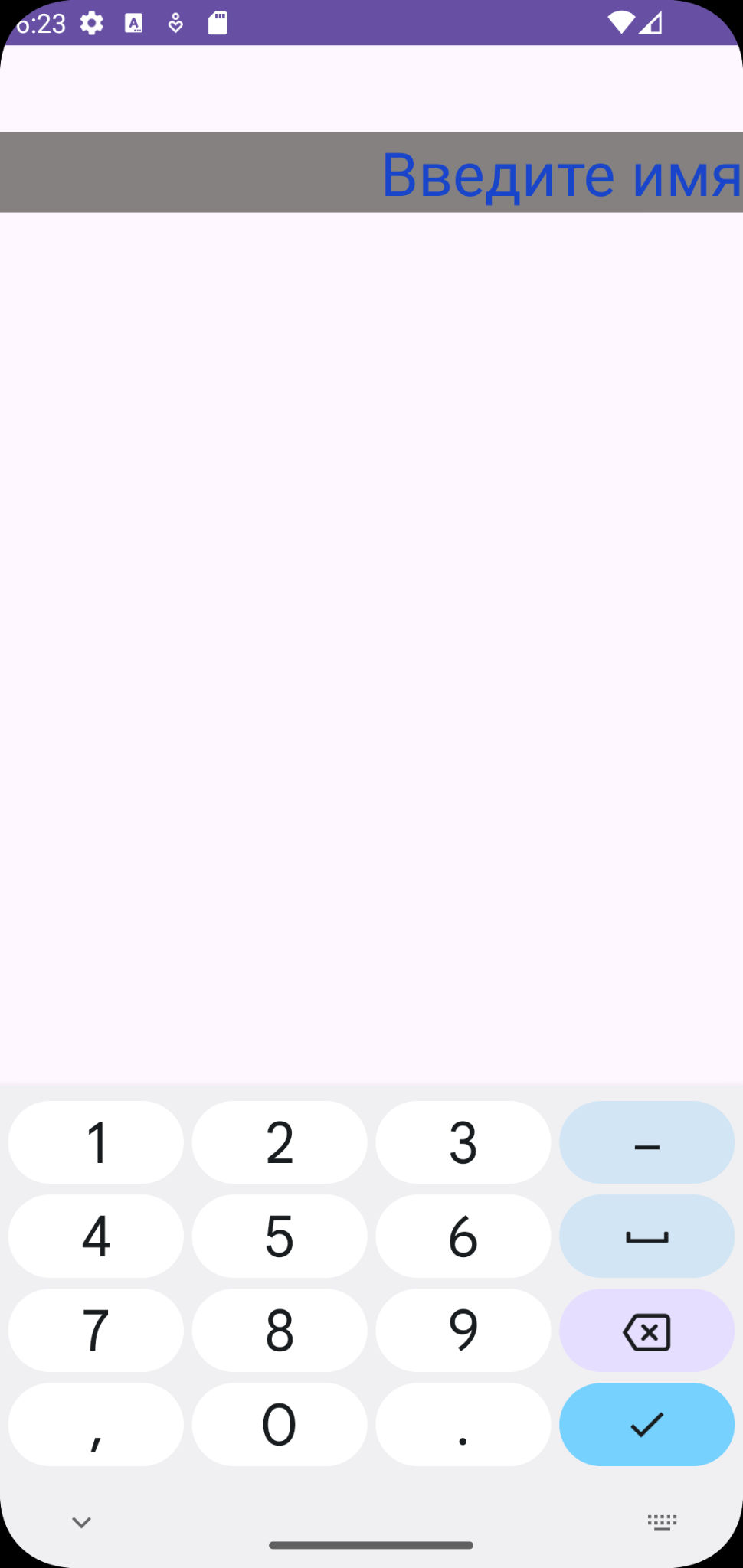




Вы можете настраивать тип разрешенных для ввода данных с помощью атрибута inputType. В зависимости от значения он устанавливает тип вводимых данных, тип экранной клавиатуры а так же то как отображаются введенные данные. Приведем примеры некоторых типов

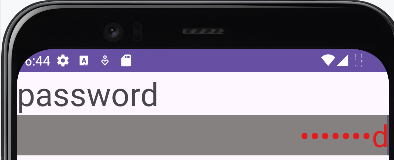
number: Только числовые значения





(text,number)Password: Заменяет введенные данные на скрывающие точки.





**Задание для самостоятельной работы:**

Составьте макет интерфейса окна авторизации входа в мобильное приложение. Тип вводимых данных должен соответствовать назначению поля для ввода текста. По нажатию кнопки приложение должно проверять заполнены ли все поля и заменить текст в верхнем текстовом поле на “Все поля заполнены корректно”

