**Лабораторная работа Android**

Создайте новый проект с пустой Activity. Имя приложения VoiceAssistent, пакет – com.example.voiceassistent. Язык программирования Java. Уровень API не ниже 21.

**Макет приложения**

Разметка файла activity\_main.xml в режиме Design должна иметь следующий вид (рис. 1).

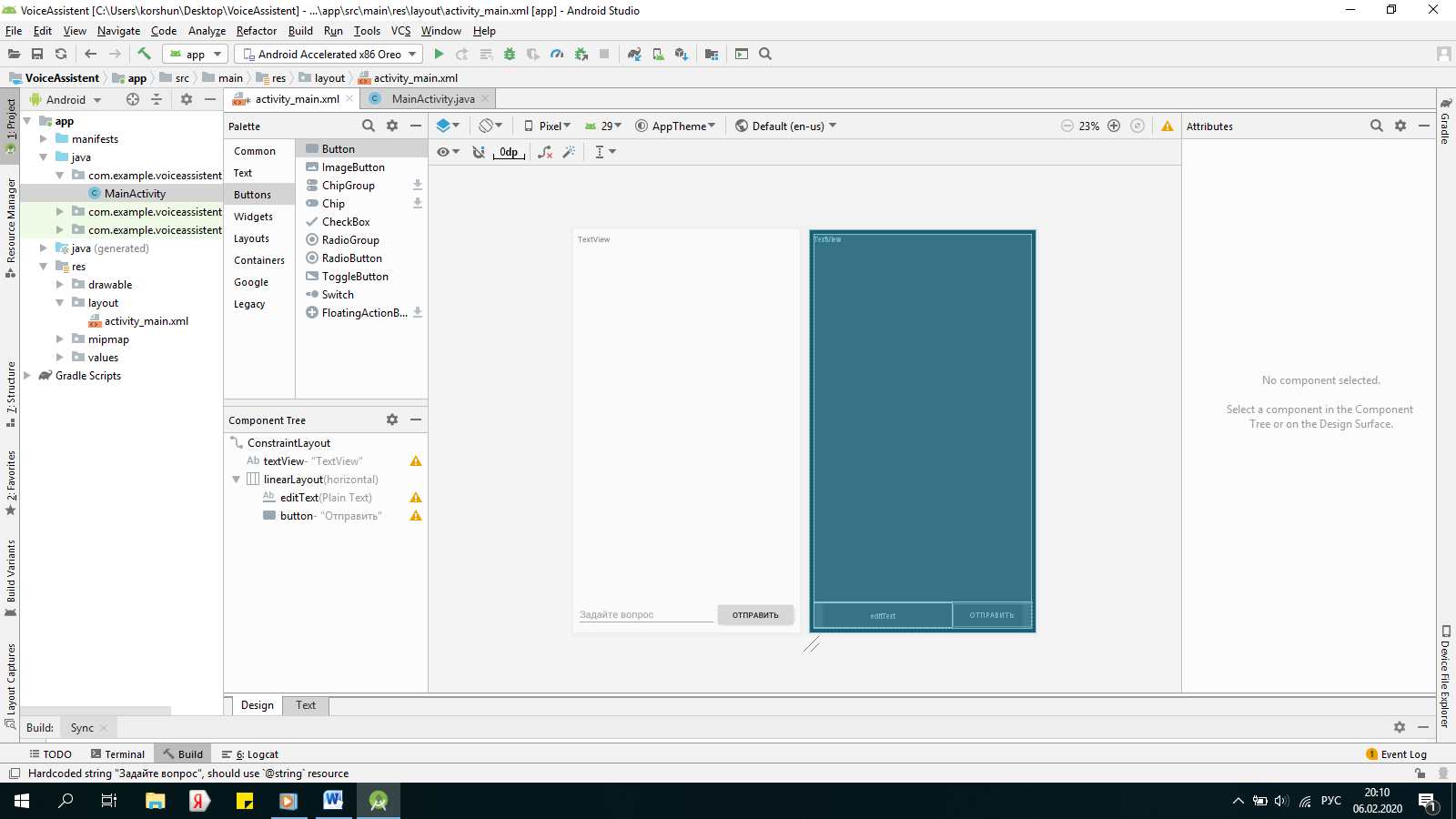


Рисунок 1. Разметка activity\_main.xml

**Работа над макетом Activity**

1. Измените свойства текстового поля (TextView)
   1. Ширина – **match\_parent**,
   2. Прикрепите к левому, правому и верхнему краям,
   3. Отступы слева, справа и сверху по **8dp**,
   4. id – **chatWindow**,
   5. text – **""**,
   6. textAlignment – **gravity**.
2. Добавьте LinearLayout (horizontal)
   1. Ширина – **match\_constraint (0dp)**,
   2. Прикрепите к левому, правому и нижнему краям,
   3. Отступы слева, справа и снизу по **8dp**.
3. Поместите внутрь LinearLayout два компонента: Plain Text (id – **questionField**, text – **""**, hint**– Задайте вопрос**) и Button (id – **sendButton**, text – **Отправить**).
4. Измените высоту LinearLayout на **wrap\_content**.
5. Прикрепите нижний край TextView к верхнему краю LinearLayout.
6. Сделайте TextView значение высоты равным **match\_constraint (0dp)**

Получившийся результат можно смело назвать «резиновым», так как он не зависит от конкретных числовых значений, зависит только от содержимого.

**Программирование**

В классе MainActivity (файле MainActivity.java) объявите следующие переменные (поля):

**protected** Button **sendButton**;  
**protected** EditText **questionText**;  
**protected** TextView **chatWindow**;

В методе onCreate «привяжите» созданные поля к соответствующим элементам:

**sendButton** = findViewById(R.id.***sendButton***);  
**questionText** = findViewById(R.id.***questionField***);  
**chatWindow** = findViewById(R.id.***chatWindow***);

При нажатии на кнопку приложение должно будет обработать вопрос и найти на него ответ. Вопрос и ответ нужно отобразить на элементе **chatWindow**. Будем обрабатывать нажатие обычным для java способом – добавим кнопке слушателя.

**Совет**. **Не копируйте этот код.** Наберите руками. Студия подскажет. Например, после new можно набирать ClickListener, студия подберет нужный класс сама и вставит его вместе с содержимым (когда нужный класс будет выбран).

**sendButton**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
   
 }  
});

Обработку нажатия нужно писать внутри метода onClick. Чтобы не загромождать метод onCreate логикой обработки, вынесем ее в отдельный метод.

**protected void** onSend() {  
  
}

Вызов данного метода поместите в onClick.

В методе onSend получите строку вопроса:

String text = **questionText**.getText().toString();  
String answer = **"Вопрос понял. Думаю…"**;

Пока ответ будет один и тот же для всех вопросов. Разнообразие ответов будет реализовано позже.

Чтобы добавить вопрос и ответ в TextView используйте метод append. Пример того как текс может появиться на экране:

>> Вопрос

<< Вопрос понял. Думаю…

Очистите поле ввода.

Запустите приложение и убедитесь в его работоспособности.

**Задание**.

1. Увеличьте размер шрифта.
2. Найдите свойство TextView, при котором текст будет появляться (добавляться) не сверху вниз, а наоборот, снизу вверх.
3. Правилом хорошего тона считается хранить ресурсы в соответствующих ресурсных файлах. Сохраните все ресурсы (строки и размеры) в соответствующих файлах.
4. Сделать локализацию.

В пакете com.example.voiceassistent создайте java-класс с именем AI. В классе создайте статический метод getAnswer, который будет принимать строку – вопрос и выдавать строку – ответ. Пусть пока это метод возвращает строку **"Вопрос понял. Думаю..."**.

Тогда вместо присваивания строкового значения нужно вызвать getAnswer для получения ответа в методе onSend класса MainActivity .

В классе AI задайте обработку основных фраз:

«Привет», ответ «Привет»

«Как дела?», ответ «Не плохо»

«Чем занимаешься?», ответ «Отвечаю на вопросы»

**Задание**.

1. Сделайте класс AI не чувствительным к регистру.
2. Сделайте так, чтобы помощник адекватно отвечал на вопросы вида: «А чем занимаешься?», «Приветик» и т.д.
3. Оформите хранение вопросов и ответов в классе AI в виде словаря (в Java это Map).

Подключить Android Studio и Genymotion. Подключаем эмулятор андроида

<https://www.youtube.com/watch?v=olP5Ohr8NqU>

<https://developer.android.com/training/basics/firstapp/building-ui>

<https://developer.android.com/studio/write/layout-editor>