#### Оценка 3:

* Реализована конвертация из метров в мм, см, м, км.
* Интерфейс соответствует скриншоту.
* Компоненты имеют осмысленные ID: txtFrom, labTo, spinnerFrom, spinnerTo, buttonConvert.

#### Оценка 4:

* Текстовые надписи вынесены в strings.xml.
* Обработка ошибок:
  + Пустое поле: "Поле не может быть пустым!".
  + Некорректное число: "Введите корректное число!".
  + Отрицательное значение: "Отрицательные значения не допускаются!".
* Добавлен второй Spinner (spinnerFrom) для выбора исходной единицы измерения.
* Ошибки выводятся в поле результата (labTo).

#### Оценка 5:

* Добавлена конвертация единиц информации: бит, байт, КБ, МБ, ГБ.
* Сохранена конвертация единиц длины.
* **Для переключения между режимами (длина/информация) используется флаг isLengthMode. В текущей реализации переключение режимов можно сделать через код**

#### Дополнительные функции:

* Форматирование результата: используется String.format("%.2f", result) для отображения двух знаков после запятой.

Переходим к кодам, их просто вставить и будет счастье.

Activity\_main.xaml:

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <!-- Поле ввода и выбор единицы измерения -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="16dp"  
 android:orientation="horizontal">  
 <TextView  
 android:id="@+id/textViewFrom"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/infoFrom"  
 android:textSize="18sp" />  
 <EditText  
 android:id="@+id/txtFrom"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:hint="@string/inputHint"  
 android:inputType="numberDecimal"  
 android:textSize="18sp" />  
 <Spinner  
 android:id="@+id/spinnerFrom"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content" />  
 </LinearLayout>  
  
 <!-- Поле результата и выбор единицы измерения -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="16dp"  
 android:orientation="horizontal">  
 <TextView  
 android:id="@+id/textViewTo"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/infoTo"  
 android:textSize="18sp" />  
 <TextView  
 android:id="@+id/labTo"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text=""  
 android:textSize="18sp" />  
 <Spinner  
 android:id="@+id/spinnerTo"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content" />  
 </LinearLayout>  
  
 <!-- Кнопка конвертации -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="16dp"  
 android:orientation="horizontal">  
 <Button  
 android:id="@+id/buttonConvert"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/buttonConvert" />  
 </LinearLayout>  
  
 <!-- Кнопка переключения режима -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <Button  
 android:id="@+id/buttonToggleMode"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/toggleMode" />  
 </LinearLayout>  
  
</LinearLayout>

String.xaml:

<resources>  
 <string name="app\_name">Конвертер Единиц</string>  
 <string name="infoFrom">От:</string>  
 <string name="infoTo">В: </string>  
 <string name="buttonConvert">Конвертировать</string>  
 <string name="toggleMode">Переключить режим</string>  
 <string name="inputHint">Введите значение</string>  
 <string name="errorEmpty">Поле не может быть пустым!</string>  
 <string name="errorInvalid">Введите корректное число!</string>  
 <string name="errorNegative">Отрицательные значения не допускаются!</string>  
 <string name="errorTooLarge">Слишком большое значение!</string>  
</resources>

Создать res/values/arrays.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <!-- Единицы измерения длины -->  
 <string-array name="length\_units">  
 <item>mm</item>  
 <item>cm</item>  
 <item>m</item>  
 <item>km</item>  
 </string-array>  
  
 <!-- Единицы измерения информации -->  
 <string-array name="data\_units">  
 <item>bit</item>  
 <item>byte</item>  
 <item>KB</item>  
 <item>MB</item>  
 <item>GB</item>  
 </string-array>  
</resources>

MainActivity.java

package com.example.unitconverter;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Spinner;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.HashMap;  
import java.util.Map;  
  
public class MainActivity extends AppCompatActivity {  
  
 private EditText txtFrom;  
 private TextView labTo;  
 private Spinner spinnerFrom, spinnerTo;  
 private boolean isLengthMode = false;  
  
 private static final Map<String, Double> *lengthToMeters* = new HashMap<>();  
 private static final Map<String, Double> *dataToBits* = new HashMap<>();  
  
 static {  
 *lengthToMeters*.put("m", 1.0);  
 *lengthToMeters*.put("mm", 0.001);  
 *lengthToMeters*.put("cm", 0.01);  
 *lengthToMeters*.put("km", 1000.0);  
  
 *dataToBits*.put("bit", 1.0);  
 *dataToBits*.put("byte", 8.0);  
 *dataToBits*.put("KB", 8.0 \* 1024);  
 *dataToBits*.put("MB", 8.0 \* 1024 \* 1024);  
 *dataToBits*.put("GB", 8.0 \* 1024 \* 1024 \* 1024);  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 txtFrom = findViewById(R.id.*txtFrom*);  
 labTo = findViewById(R.id.*labTo*);  
 spinnerFrom = findViewById(R.id.*spinnerFrom*);  
 spinnerTo = findViewById(R.id.*spinnerTo*);  
 Button buttonConvert = findViewById(R.id.*buttonConvert*);  
 Button buttonToggleMode = findViewById(R.id.*buttonToggleMode*);  
  
 updateSpinners();  
  
 buttonConvert.setOnClickListener(v -> performConversion());  
 buttonToggleMode.setOnClickListener(v -> toggleMode(v));  
  
 spinnerFrom.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  
 labTo.setText("");  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> parent) {}  
 });  
  
 spinnerTo.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  
 labTo.setText("");  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> parent) {}  
 });  
 }  
  
 private void updateSpinners() {  
 ArrayAdapter<CharSequence> adapter;  
 if (isLengthMode) {  
 adapter = ArrayAdapter.*createFromResource*(this,  
 R.array.*length\_units*, android.R.layout.*simple\_spinner\_item*);  
 } else {  
 adapter = ArrayAdapter.*createFromResource*(this,  
 R.array.*data\_units*, android.R.layout.*simple\_spinner\_item*);  
 }  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
 spinnerFrom.setAdapter(adapter);  
 spinnerTo.setAdapter(adapter);  
 }  
  
 private void performConversion() {  
 String inputText = txtFrom.getText().toString();  
  
 if (inputText.isEmpty()) {  
 labTo.setText(getString(R.string.*errorEmpty*));  
 return;  
 }  
  
 try {  
 double value = Double.*parseDouble*(inputText);  
  
 if (value < 0) {  
 labTo.setText(getString(R.string.*errorNegative*));  
 return;  
 }  
  
 if (value > 1\_000\_000\_000) {  
 labTo.setText(getString(R.string.*errorTooLarge*));  
 return;  
 }  
  
 double result;  
 if (isLengthMode) {  
 result = convertLength(value, spinnerFrom.getSelectedItem().toString(),  
 spinnerTo.getSelectedItem().toString());  
 } else {  
 result = convertData(value, spinnerFrom.getSelectedItem().toString(),  
 spinnerTo.getSelectedItem().toString());  
 }  
  
 if (result % 1 == 0) {  
 labTo.setText(String.*format*("%,d", (long) result));  
 } else {  
 labTo.setText(String.*format*("%,.2f", result));  
 }  
 } catch (NumberFormatException e) {  
 labTo.setText(getString(R.string.*errorInvalid*));  
 }  
 }  
  
 private double convertLength(double value, String fromUnit, String toUnit) {  
 double inMeters = value \* *lengthToMeters*.getOrDefault(fromUnit, 1.0);  
 return inMeters / *lengthToMeters*.getOrDefault(toUnit, 1.0);  
 }  
  
 private double convertData(double value, String fromUnit, String toUnit) {  
 double inBits = value \* *dataToBits*.getOrDefault(fromUnit, 1.0);  
 return inBits / *dataToBits*.getOrDefault(toUnit, 1.0);  
 }  
  
 public void toggleMode(View view) {  
 isLengthMode = !isLengthMode;  
 updateSpinners();  
 labTo.setText("");  
 txtFrom.setText("");  
 }  
}