|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИИТ)**

**Кафедра Математического обеспечения и стандартизации информационных технологий**

Отчет по выполнению практического задания №2

**Тема:**

**Основы визуального конструирования**

Дисциплина:  
Разработка мобильных приложений

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил:  Студент группы ИКБО-03-20 |  |  | Цемкало А.Р. |
|  |  |  |  |
| Принял:  Доцент кафедры МОСИТ ИИТ |  |  | Чернов Е. А. |

Москва 2022 г.

СОДЕРЖАНИЕ

[Цель работы 3](#_Toc96380500)

[Ход работы 3](#_Toc96380501)

[A. Приёмы визуального конструирования содержимого экрана. 3](#_Toc96380502)

[Б. Индивидуальное задание. 12](#_Toc96380503)

[Вывод 20](#_Toc96380504)

Цель работы

Познакомиться с начальными приёмами визуального конструирования на примере простой программы.

Ход работы

1. Приёмы визуального конструирования содержимого экрана.
2. Создание нового проекта: выбор шаблона Activity («Empty Activity»), выбор имени проекта, языка программирования, выбор минимальной версии SDK.

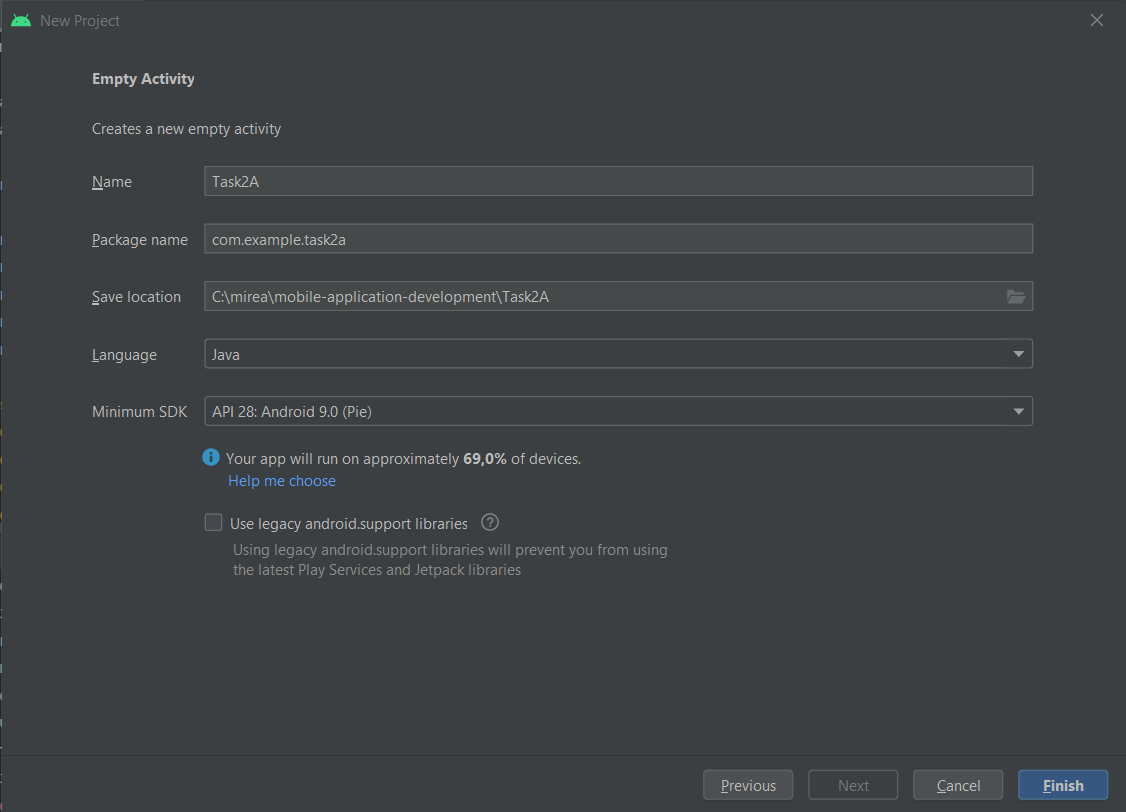


Рисунок 1 – скриншот создания нового проекта

Редактирование файла activity\_main.xml: добавление текста и кнопки.

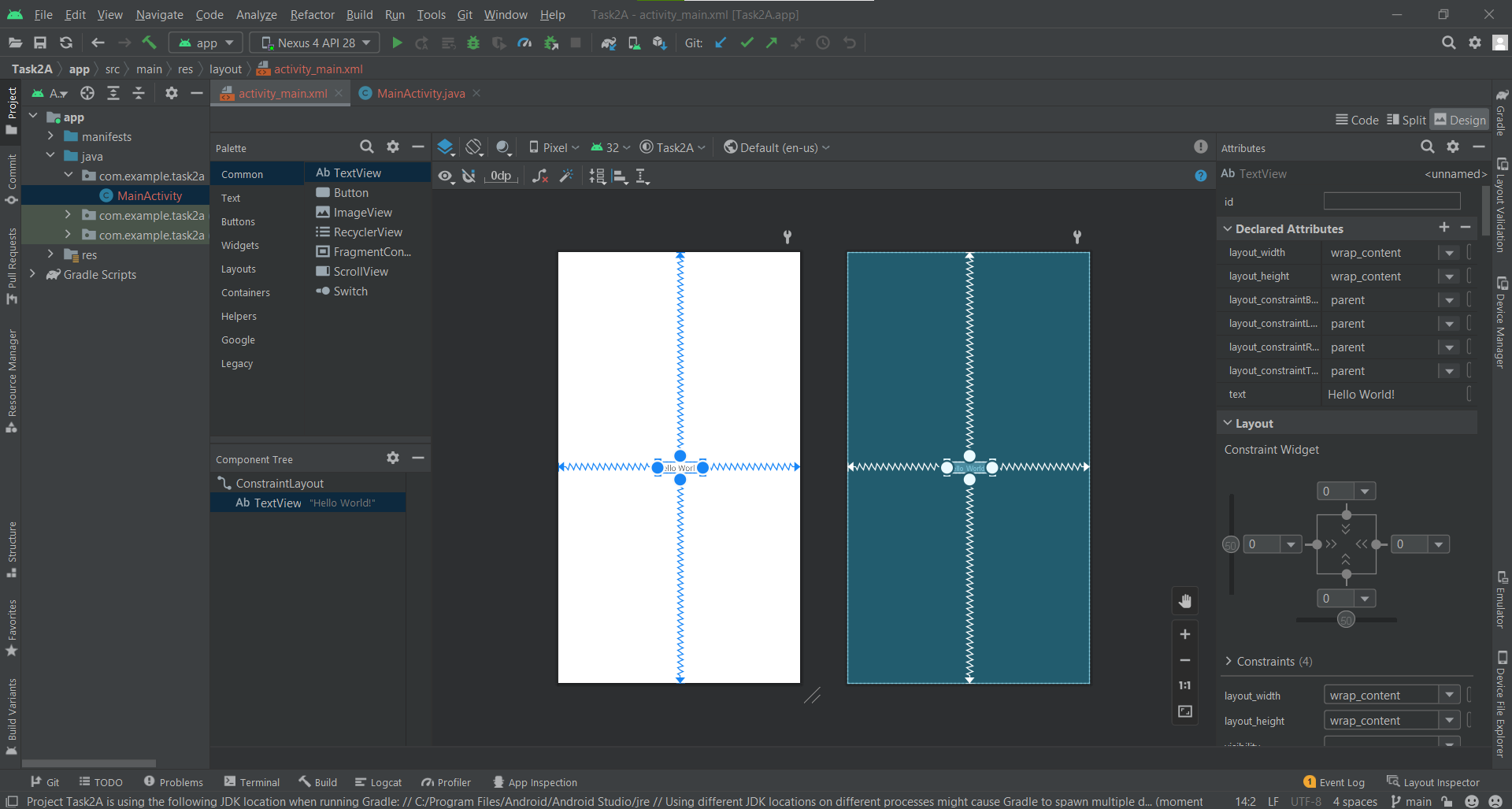


Рисунок 2 – Графическое представление Activity (вкладка Design)

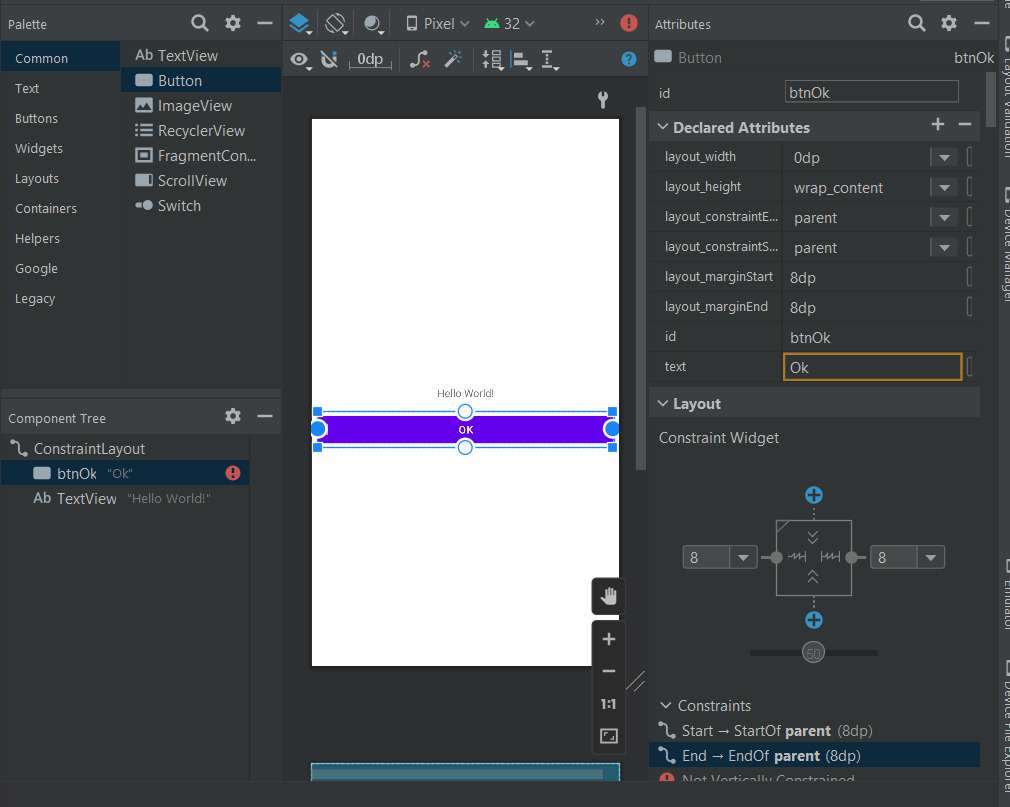


Рисунок 3 – привязка кнопки к боковым краям корневого контейнера

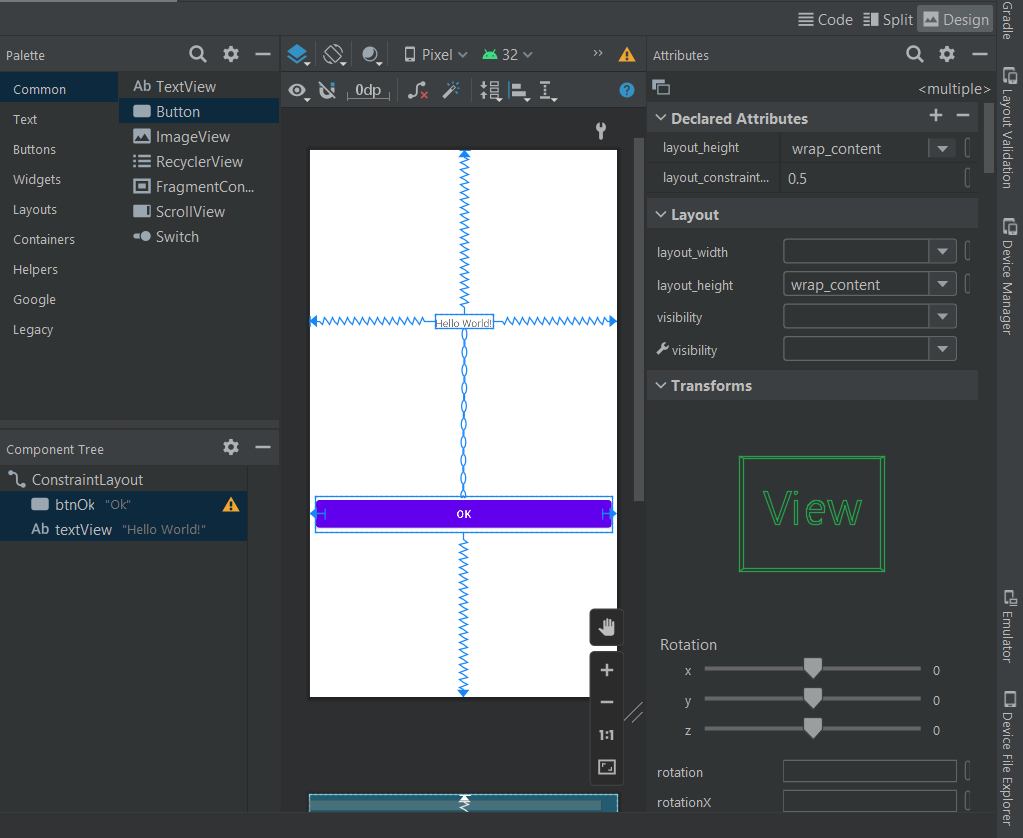


Рисунок 4 – объединение текстового поля и кнопки в вертикальную цепь

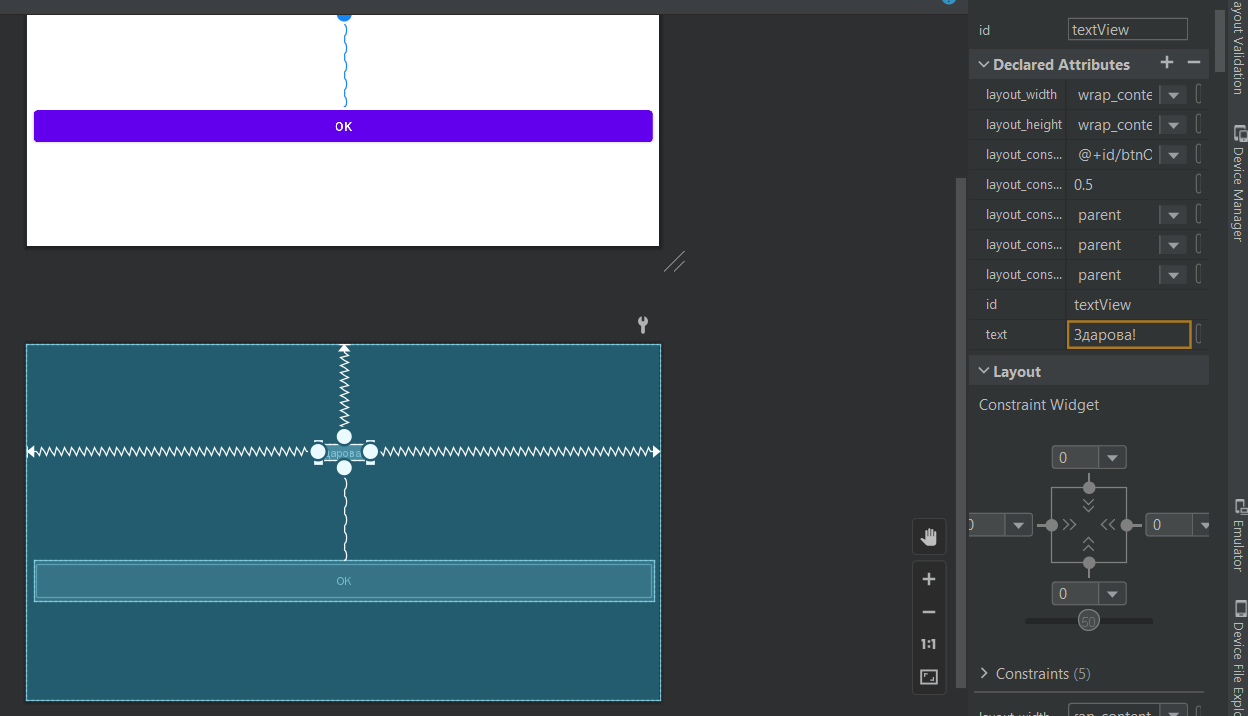


Рисунок 5 – Создание графического представления для горизонтальной ориентации экрана.

1. Запуск приложения.

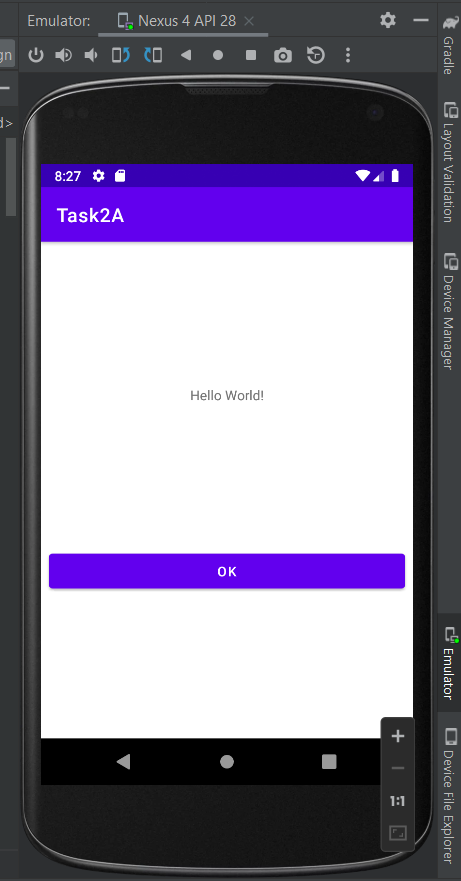


Рисунок 6 – Вид работающего приложения в виртуальном девайсе в вертикальном положении

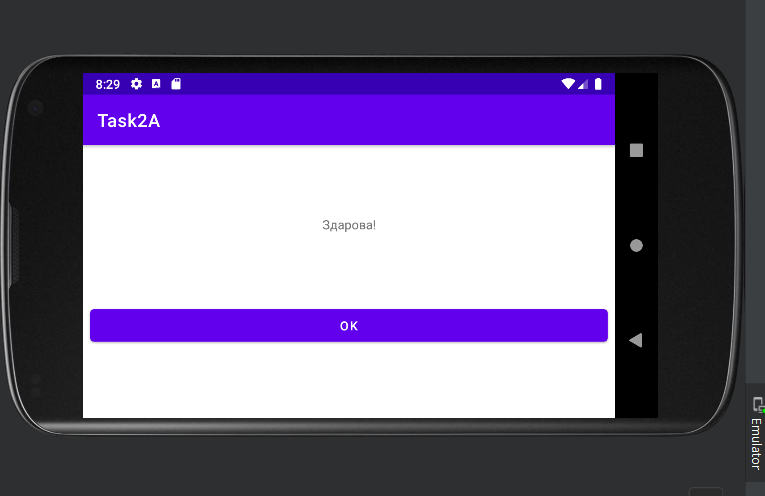


Рисунок 7 – Вид работающего приложения в виртуальном девайсе в горизонтальном положении

1. Доступ к элементам экрана из кода.

Код файла MainActivity.java

package com.example.task2a;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.Button;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

private Button btnOk;

private TextView textView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

initViews();

}

private void initViews() {

btnOk = (Button) findViewById(R.id.btnOk);

textView = (TextView) findViewById(R.id.textView);

if (btnOk != null)

textView.setText("Кнопка есть!");

else textView.setText("Кнопки нет!");

}

}

1. Проверка работы файла MainActivity.java

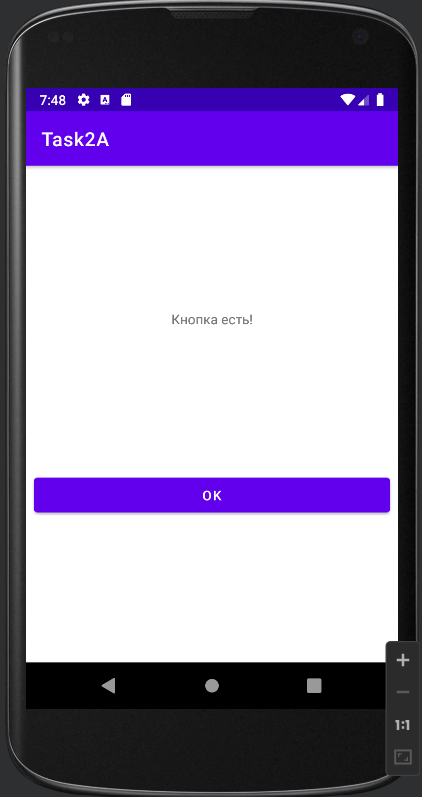


Рисунок 8 – Вид работающего приложения в виртуальном девайсе в вертикальном положении

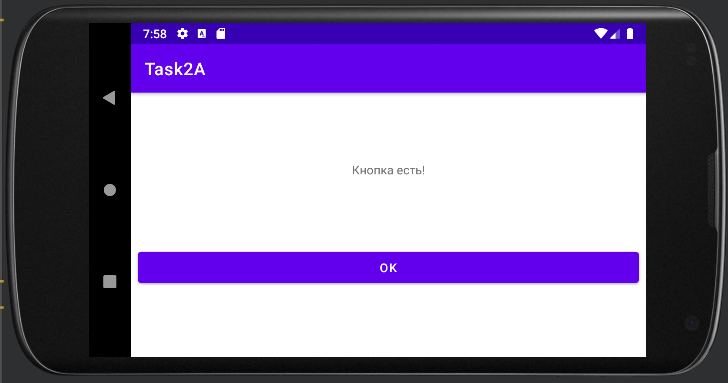


Рисунок 9 – Вид работающего приложения в виртуальном девайсе в горизонтальном положении

В обоих положениях в textView отображается надпись «Кнопка есть!», так как содержимое поля изменено java кодом.

1. Обработка событий на примере нажатия кнопки

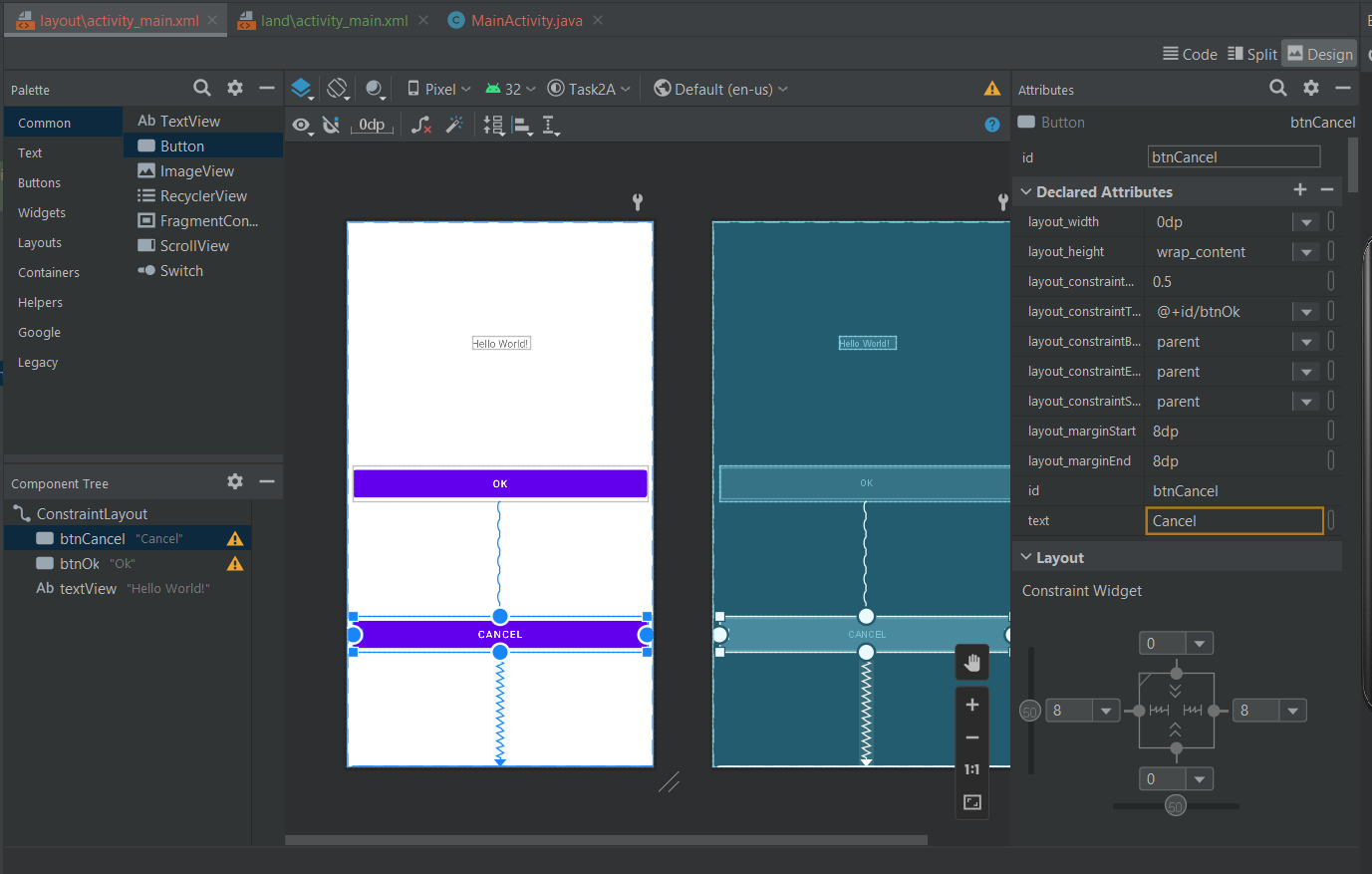


Рисунок 10 – Настройка кнопки Cancel для вертикально положения экрана

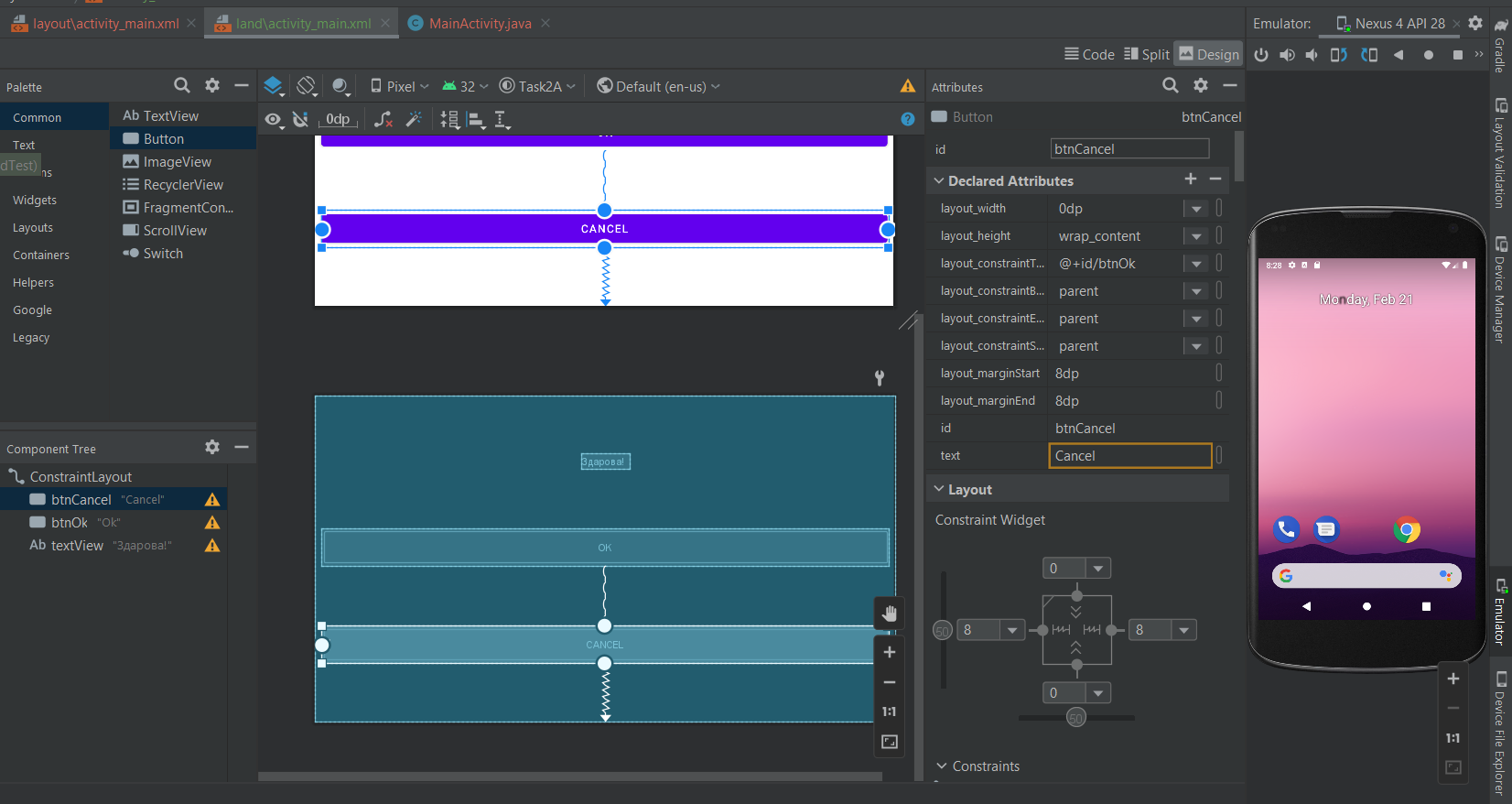


Рисунок 11 – Настройка кнопки Cancel для горизонтального положения экрана

Код MainActivity.java

package com.example.task2a;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

private Button btnOk;

private Button btnCancel;

private TextView textView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

initViews();

}

private void initViews() {

btnOk = (Button) findViewById(R.id.btnOk);

btnCancel = (Button) findViewById(R.id.btnCancel);

textView = (TextView) findViewById(R.id.textView);

textView.setText("Нажми на любую кнопку!");

btnOk.setOnClickListener(this);

btnCancel.setOnClickListener(this);

if (btnOk != null)

textView.setText("Кнопка есть!");

else textView.setText("Кнопки нет!");

}

@Override

public void onClick(View view) {

switch (view.getId()) {

case R.id.btnOk:

textView.setText("Ok! Нажми Cancel");

btnOk.setEnabled(false);

btnCancel.setEnabled(true);

break;

case R.id.btnCancel:

textView.setText("Cancel! Нажми Ok");

btnCancel.setEnabled(false);

btnOk.setEnabled(true);

break;

}

}

}

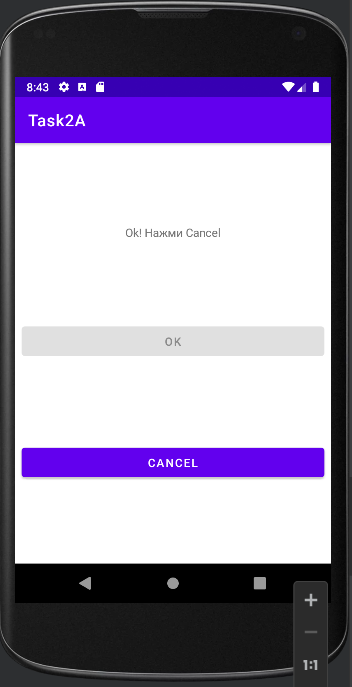


Рисунок 10 – Вид работающего приложения при нажатой кнопке Ok

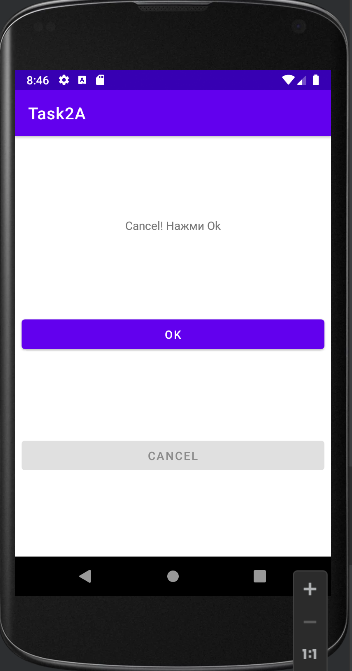


Рисунок 11 – Вид работающего приложения при нажатой кнопке Cancel

Б. Индивидуальное задание.

Вариант 6.

IPv4-калькулятор, который по заданным в десятичном виде IP-адресу и маске подсети определяет длину маски, диапазон допустимых адресов, адрес подсети, адрес широковещательной рассылки в этой подсети.

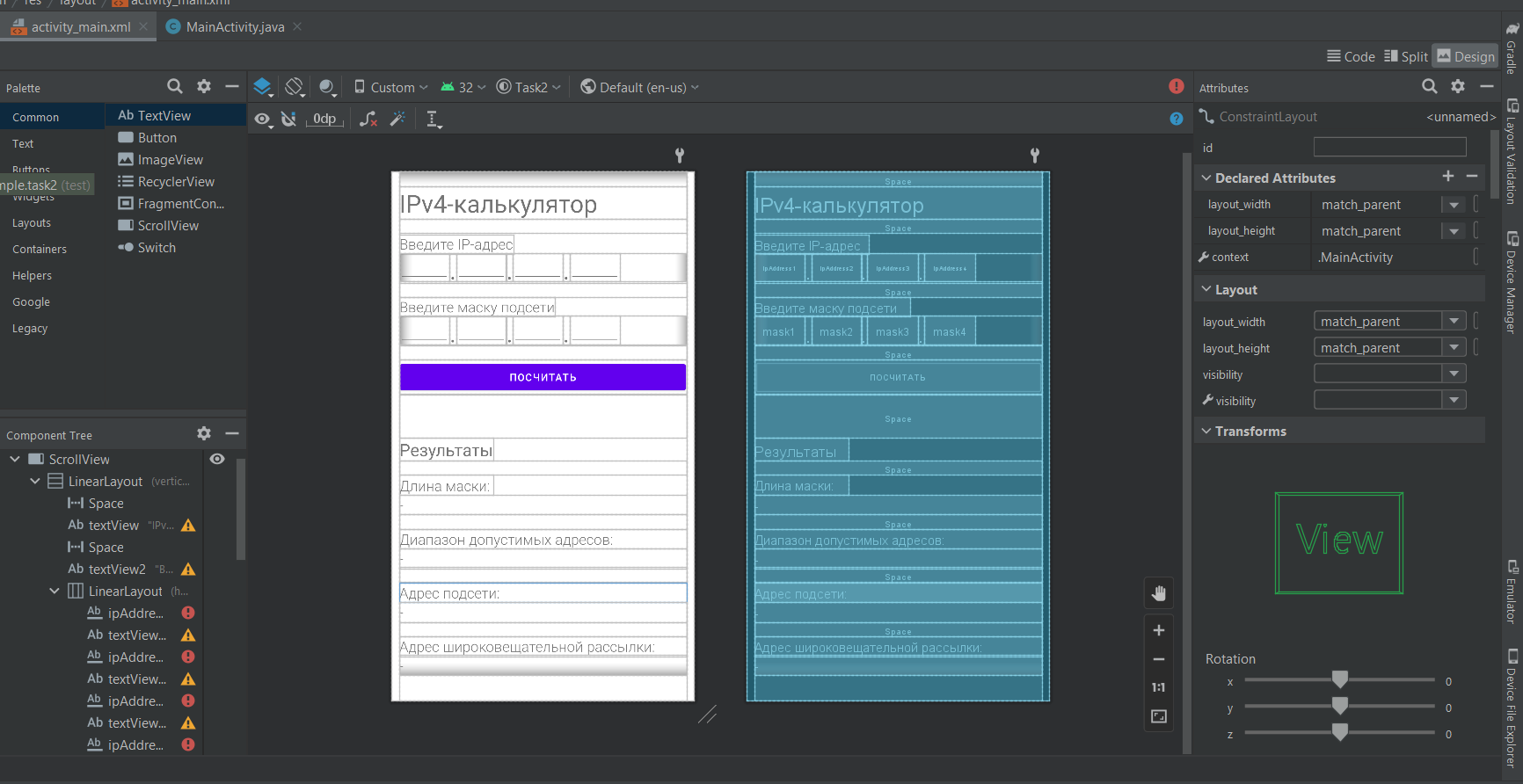


Рисунок 12 – Графическое представление Activity (вкладка Design)

Код MainActivity.java

package com.example.task2;

import java.lang.\*;

import java.util.Collections;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

private Button btnCount;

private TextView textMaskLength;

private TextView textAddressRange;

private TextView textSubnetAddress;

private TextView textBroadcastAddress;

private EditText textIpAddress1;

private EditText textIpAddress2;

private EditText textIpAddress3;

private EditText textIpAddress4;

private EditText textMask1;

private EditText textMask2;

private EditText textMask3;

private EditText textMask4;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

initViews();

}

private void initViews() {

btnCount = (Button) findViewById(R.id.btnCount);

textMaskLength = (TextView) findViewById(R.id.maskLength);

textAddressRange = (TextView) findViewById(R.id.addressRange);

textSubnetAddress = (TextView) findViewById(R.id.subnetAddress);

textBroadcastAddress = (TextView) findViewById(R.id.broadcastAddress);

textIpAddress1 = (EditText) findViewById(R.id.ipAddress1);

textIpAddress2 = (EditText) findViewById(R.id.ipAddress2);

textIpAddress3 = (EditText) findViewById(R.id.ipAddress3);

textIpAddress4 = (EditText) findViewById(R.id.ipAddress4);

textMask1 = (EditText) findViewById(R.id.mask1);

textMask2 = (EditText) findViewById(R.id.mask2);

textMask3 = (EditText) findViewById(R.id.mask3);

textMask4 = (EditText) findViewById(R.id.mask4);

btnCount.setOnClickListener(this);

}

@Override

public void onClick(View v) {

switch (v.getId()) {

case R.id.btnCount:

Integer ip1 = Integer.parseInt(textIpAddress1.getText().toString());

Integer ip2 = Integer.parseInt(textIpAddress2.getText().toString());

Integer ip3 = Integer.parseInt(textIpAddress3.getText().toString());

Integer ip4 = Integer.parseInt(textIpAddress4.getText().toString());

Integer mask1 = Integer.parseInt(textMask1.getText().toString());

Integer mask2 = Integer.parseInt(textMask2.getText().toString());

Integer mask3 = Integer.parseInt(textMask3.getText().toString());

Integer mask4 = Integer.parseInt(textMask4.getText().toString());

// Integer mask1 = 255;

// Integer mask2 = 255;

// Integer mask3 = 248;

// Integer mask4 = 0;

// Integer ip1 = 192;

// Integer ip2 = 168;

// Integer ip3 = 11;

// Integer ip4 = 10;

Integer lengthMask = 0;

Integer[] arrayLengthMask = {countMaskLength(mask1), countMaskLength(mask2), countMaskLength(mask3), countMaskLength(mask4)};

for (int i = 0; i < 4; i++) {

if (arrayLengthMask[i] == -1) {

textMaskLength.setText("Invalid mask");

break;

}

lengthMask += arrayLengthMask[i];

}

textMaskLength.setText(lengthMask.toString());

String[] binarySubnetAddressArray = {getEightBytes(ip1 & mask1), getEightBytes(ip2 & mask2), getEightBytes(ip3 & mask3), getEightBytes(ip4 & mask4)};

textSubnetAddress.setText(String.join(".", binarySubnetAddressArray));

textAddressRange.setText(countAddressRange(String.join("", binarySubnetAddressArray)).toString());

textBroadcastAddress.setText(getBroadcastAddress(new Integer[]{ip1, ip2, ip3, ip4}, new Integer[]{mask1, mask2, mask3, mask4}));

break;

}

}

public static Integer countMaskLength(Integer mask) {

if (mask == 0) {

return 0;

} else if (mask == 255) {

return 8;

}

String binaryMask = Integer.toBinaryString(mask);

if (binaryMask.length() != 8) {

return -1;

}

boolean zeroFound = false;

Integer k = 0;

for (int i = 0; i < 8; i++) {

if (binaryMask.charAt(i) == '1') {

if (zeroFound) {

return -1;

}

k++;

} else {

zeroFound = true;

}

}

return k;

}

public static String getEightBytes(Integer x) {

String newStr = Integer.toBinaryString(x);

if (newStr.length() == 8) {

return newStr;

}

return String.join("", Collections.nCopies(8 - newStr.length(), "0")) + newStr;

}

public static Integer countAddressRange(String address) {

int i;

for (i = address.length() - 1; i >= 0; i--) {

if (address.charAt(i) == '1') {

break;

}

}

Integer zeros = address.length() - i - 1;

return (int) Math.pow(2, zeros) - 2;

}

public static String getBroadcastAddress(Integer[] ip, Integer[] mask) {

String[] broadcastAddress = new String[4];

for (int i = 0; i < 4; i++) {

broadcastAddress[i] = Integer.toString(ip[i] | (~(mask[i]) + 256));

}

return String.join(".", broadcastAddress);

}

}

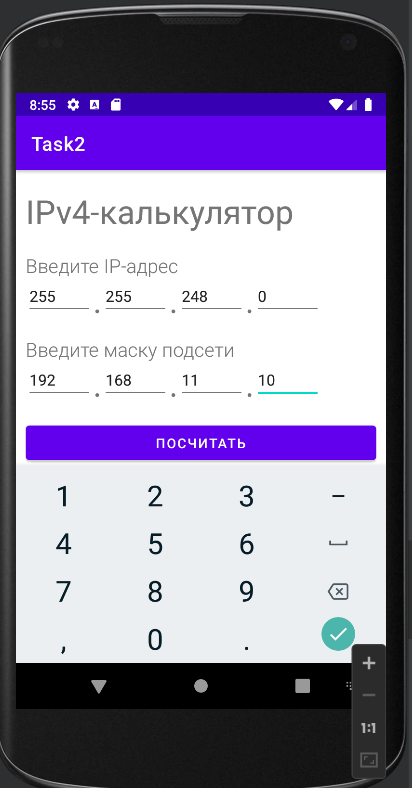


Рисунок 13 – Ввод значений в приложение

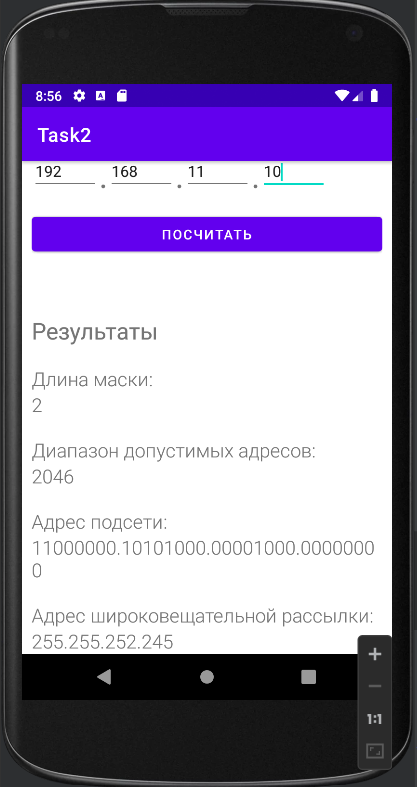


Рисунок 14 – Вид приложения после ввода значений и нажатия на кнопку для подсчёта результатов

Вывод

Освоены начальные приёмы визуального конструирования на примере простой программы.