

Flutter: Android Studioda oddiy kalkulyator dasturini yaratish, uni emulyatorda sinovdan o'tkazish va mobil telefonga o'rnatib ishlatish

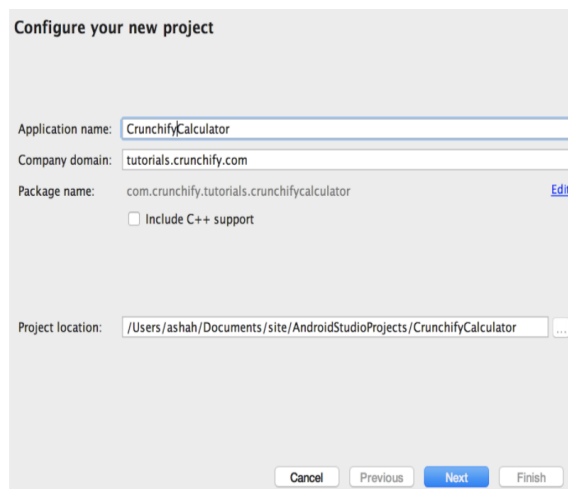
Reja:

1. Android Studioda oddiy kalkulyator dasturini yaratish
2. Kalkulyatorni emulyatordan sinovdan o'tkazish va mobil telefonga o'rnatib ishlatish

Android studioda kalkulyator dasturini yaratish quyidagi bosqichlarda amalga oshiriladi

1- bosqichi. Android Studioni oching. Yangi Android Studio loyihasini ishga tushirishni bosing.

Loyihaga nom bering. Kalkulator qolgan joylari o'z holicha qoladi.



2-bosqichi

Minimal SDK API 15: Android 4.0.3(IceCreamSandwich)ni tanlaymiz , masalan, API 15 (IceCreamSandwich), chunki u qurilmalarning deyarli 94% ni qamrab oladi va deyarli barcha xususiyatlarga ega. Agar 100% qurilmani qamrab olishni istasak, API 8 ni tanlashimiz mumkin: Android 2.2 (Froyo).

☒ Phone and Tablet

Minimum SDK **API 15: Android 4.0.3 (IceCreamSandwich)**

Lower API levels target more devices, but have fewer features available.

By targeting API 15 and later, your app will run on approximately **100.0%** of the devices that are active on the Google Play Store.

[Help me choose](#)

3-qadam

Empty Activity ni tanlaymiz va next tugmasini bosamiz. Qolgan joylarining o'zi qoladi.

Creates a new empty activity

Activity Name: **MainActivity**

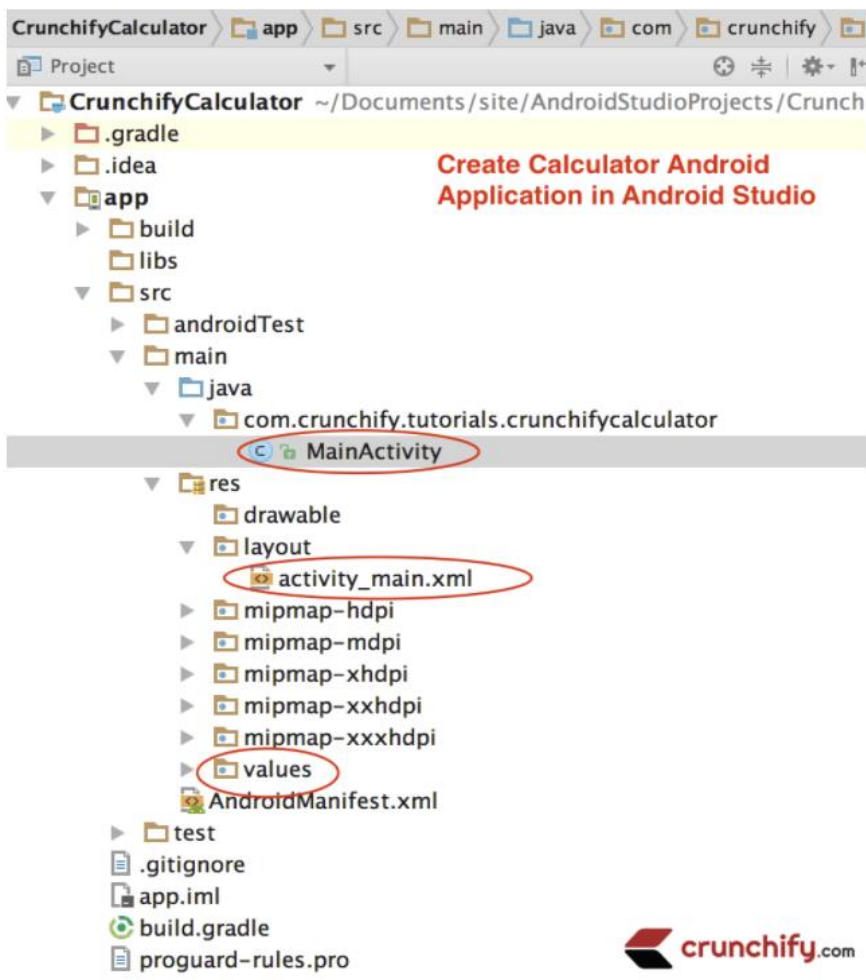
☒ Generate Layout File

Layout Name: **activity_main**

☒ Backwards Compatibility (AppCompat)

4 bosqichi

Loyihamizni yaratish uchun "tayyor" tugmasini bosgandan so'ng, loyihamiz oynasi ochiladi.



Kalkulyator ilovasida to'rtta asosiy element mavjud:

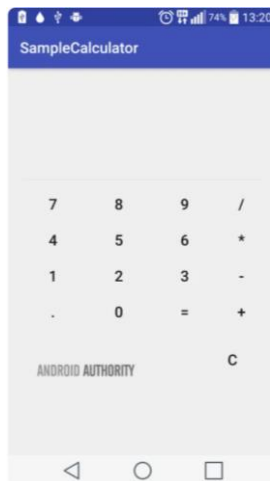
RelativeLayout - ekranda boshqa elementlar qanday stacked yoki ko'rsatilishini aniqlaydi. RelativeLayout ichidagi elementlarni bir-biriga yoki o'zlariga nisbatan joylashtirish uchun ishlatiladi.

TextView - element matnni ko'rsatish uchun ishlatiladi. Yozilgan ma'lumotni tahrirlab bo'lmaydi. Hisoblash natijalari TextView yordamida ko'rsatiladi.

EditText - TextView boshqaruviga o'xshash yagona farq shundaki, foydalanuvchilar matnni tahrirlashlari mumkin. Ammo kalkulyator faqat kiritilgan ma'lumotlarning belgilangan to'plamiga ruxsat berganligi sababli, biz uning holatini "tahrirlab bo'lmaydi" deb belgilaymiz. Foydalanuvchi raqamlarni bosganda, biz ularni EditTextga chiqaramiz.

Tugma - foydalanuvchi buyruqlariga javob beradi. Oddiy Android ilovasini yaratishda biz kalkulyatorda raqamlar va harakat operatorlari tugmalaridan foydalanamiz.

Kalkulyator ko'rinishi



Kalkulyatorning ichki komponentlari

Android telefonida ilovani yaratishdan oldin, "value One" va "valueTwo" da ishlatiladigan raqamlar mavjudligini ta'kidlaymiz. Har ikki o'zgaruvchilar uchun ham double turi mavjuddir. Biz Valueone uchun maxsus NaN qiymatini o'rnatamiz. Quidagicha yoziladi:

```
private double valueOne = Double.NaN;
```

```
private double valueTwo;
```

Ushbu oddiy kalkulyator faqat qo'shish, ayirish, ko'paytirish va bo'lish operatsiyalarini bajarishi mumkin. Shuning uchun biz ushbu operatsiyalarni taqdim etish uchun to'rtta statik belgini va quyidagi operatsiyani o'z ichiga olgan CURRENT_ACTION o'zgaruvchisini aniqlaymiz.

```
private static final char ADDITION = '+';
```

```
private static final char SUBTRACTION = '-'; private static final char  
MULTIPLICATION = '*';
```

```
private static final char DIVISION = '/';
```

```
private char CURRENT_ACTION;
```

Keyin natijani formatlash uchun DecimalFormat sinfidan foydalanamiz.

```
decimalFormat = new DecimalFormat("#.#####");
```

Raqamlarni bosish jarayoni

Android uchun yaratilgan oddiy ilovamizda foydalanuvchi raqamni yoki nuqtani bosganda, biz ushbu raqamni editText-ga qo'shishimiz kerak.

Quyidagi kod misoli, bu raqam uchun qanday amalga oshirilganligini ko'rsatadi.

```
binding.buttonZero.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view)
    { binding.editText.setText(binding.editText.getText() + "0"); } });
```

Kalkulyator yaratish

Android Studio dasturini ishga tushiramiz, yangi loyiha yaratamiz. Komponentlarning nomlarini o'z holicha qoldiraman: **faoliyat nomi** - asosiy, fayl tartibi-**activity_main**.

Ekranda 2 ta EditText va 4 ta Button tugmasidan foydalanamiz.

Misol:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/et2"
        android:hint="@string/num"/>
    <EditText
        android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
android:id="@+id/et1"
android:hint="@string/num"/>
```

<TextView

```
android:layout_width="fill_parent"
android:layout_height="20pt"
android:id="@+id/result"
android:text="Results"/>
```

<Button

```
android:layout_width="30pt"
android:layout_height="30pt"
android:id="@+id/plus"
android:text="+"
android:textSize="20pt"/>
```

<Button

```
android:layout_width="30pt"
android:layout_height="30pt"
android:id="@+id/minus"
android:text="-"
android:textSize="20pt"/>
```

<Button

```
android:layout_width="30pt"
android:layout_height="30pt"
android:id="@+id/multiply"
android:text="*"
android:textSize="20pt"/>
```

<Button

```
android:layout_width="30pt"
android:layout_height="30pt"
```

```
android:id="@+id/divide"
android:text="/"
android:textSize="20pt"/>
```

</RelativeLayout>

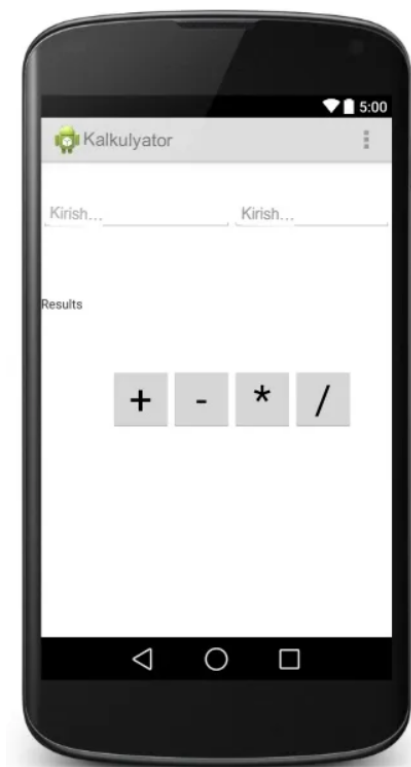
<resources>

```
<string name="app_name">My Application</string>
<string name="hello_world">Hello world!</string>>
<string name="action_settings">>chiqish</string>>
<string name="calculator">>Kalkulyator</string>
```

```
<string name="num">>kirish...</string>
```

</resources>

Barcha elementlarni yaratganingizdan so'ng, dizayn rejimiga o'tamiz va yaratilgan elementlarni biz xohlagancha harakatlantirishimiz mumkin.



Biz uchun zarur bo'lgan barcha elementlarni Yaratgandan so'ng, dasturni **MainActivity.java**. ga yozamiz.

Bizga kerak:

- 1) ishlatilgan narsalarni e'lon qilish,
- 2) ushbu moslamalarni biz yaratgan elementlarga ulash,
- 3) berilgan tugmachalar bosilganda ko'rsatish,
- 4) tugmalarining har birini bosish jarayonida nima bo'lishini tushuntiring.

Dastur kodi:

```
package home.myapplication;
import android.app.Activity;
import android.app.AlertDialog;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
public class MainActivity extends Activity implements View.OnClickListener {
// Quyidagi ob'ektlarning mavjudligini e'lon qiling:
    private EditText mE1; private EditText mE2; private TextView mResult;
    private Button mB1; private Button mB2; private Button mB3; private
    Button mB4; private String mCount = "";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); //
        Kiritilgan dasturiy moslamalarni formatlash faylida yaratilgan elementlarga
```

ulang:

```
mE1=(EditText)findViewById(R.id.et1);
mE2=(EditText)findViewById(R.id.et2);
mResult=(TextView)findViewById(R.id.result);
mB1=(Button)findViewById(R.id.plus);
mB2=(Button)findViewById(R.id.minus);
mB3=(Button)findViewById(R.id.multiply);
mB4=(Button)findViewById(R.id.divide); // Bizning tugmachalarni bosing:
mB1.setOnClickListener(this);           mB2.setOnClickListener(this);
mB3.setOnClickListener(this); mB4.setOnClickListener(this); } @Override
public void onClick(View v) {
    // Biz faoliyat ko'rsatadigan o'zgaruvchilarni kiriting:
    float value1 =0;
    float value2=0;
    float result=0;
    // Kirish elementlari bilan value1 va value2 uchun qadriyatlarni qabul:
    value1=Float.parseFloat(mE1.getText().toString());
    value2=Float.parseFloat(mE2.getText().toString()); // Tugmalar harakatlarini
    tavsiflang:
    switch (v.getId()) { case R.id.plus:
        mCount = "+";
        result = value1 + value2; break;
        case R.id.minus:
            mCount = "-";
            result = value1 - value2; break;
        case R.id.multiply:
            mCount = "*";
            result = value1 * value2; break;
        case R.id.divide:
```

```
mCount = "/";  
result = value1 / value2; break;  
default:  
break;  
}  
// natijani namoyish qilishni sozlash:  
mResult.setText(value1 + " " + mCount + " " + value2 + " = " + result); } }
```

Nazorat savollari

1. Android Studioda oddiy kalkulyator dasturini yaratish ketma-ketligini tushuntiring?
2. Kalkulyatorni emulyatordan sinovdan o'tkazish qay tartibda amalga oshiriladi?
3. Yaratilgan kalkulyatorni mobil telefonga o'rnatib ishlatish jarayoni aytib bering?