

**Міністерство освіти і науки України**  
**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра обчислювальної техніки**

**Розрахунково-графічна робота**  
з дисципліни  
«Об'єктно-орієнтоване програмування»

Виконав:

студент групи ІП-93

Домінський Валентин Олексійович

Номер залікової книжки: 9311

Номер у списку: 9

Перевірив:

Порєв Віктор Миколайович

Київ 2020

## Вступ:

За списком Моєю темою став звичайний калькулятор. Спочатку Я хотів писати на звичному для Мене C#, але в кінці передумав, з кількох причин:

1. Постійно писати на C# нудно
2. Викладач заохочує писати на інших мовах, таких як Java, Kotlin та програми під Android у цілому

Отже мовою програмування став Kotlin, оскільки ця мова молода та доволі перспективна, цільовою операційною системою став Android, а середовищем для створення – Android Studio.

## Початок роботи:

При роботі над новим проектом Я завжди використовую схему «зроби хоч якось, аби працювало». У даного методу є багато противників і хочу зізнатися, що Я теж один з них, але, все-таки, особисто Мені так легше працювати. Ось деякі плюси цього способу:

1. Доволі швидко можна отримати готовий продукт або фінальний результат.
2. При рефакторингу у Вас завжди буде працюючий код, за допомогою якого Ви зможете перевіряти правильність Своїх дій для створення нового, більш ефективного коду.

Першою новинкою, з якою ще треба було ознайомитися, стала XML-розмітка. Я ніколи не працював з даною технологією та перша година роботи була непростю, але потім Я звик до неї і вона стала виглядати доволі зручною. Також в Andoid studio є гарна функція показу коду розмітки та її вигляду для користувачів, що стало в нагоді ще не раз.

## Перша реалізація:

Як Я вже казав, користуючись методом «зроби хоч якось, аби працювало», можна доволі швидко отримати результат, що в Моєму випадку і сталося. Приблизна реалізація виглядала якось так:

1. Кожна кнопка мала на собі Listener, який викликав функцію A.1
2. Функція A.1 змінювала теперішню операцію (наприклад на «+») та передавала в іншу функцію Б, яка була спільна для багатьох функцій А
3. У функції Б, у залежності від операції та тексту, який вводив користувач, виконувалися певні операції (наприклад операція, яка працювала з двома числами або ж інша, та, що мала діло лише з одним числом), в яких уже відбувалися математичні операції, а результат передавався в один з TextView

Мінуси даного варіанту очевидні:

1. Усе знаходиться лише в одному класі
2. Кожен раз викликати лише певну функцію не давало змоги програмі стати більш гнучкою
3. Великі Switch / if, які й відповідали за логіку виконання певних операцій

*Приклад одного з умовних операторів:*

```
if (operation == '+') {  
  
    result = tempValue1.toDouble() + tempValue2.toDouble()  
  
} else if (operation == '-') {  
    result = tempValue1.toDouble() - tempValue2.toDouble()  
}
```

І це лише дві операції! А у Мене таких велика кількість! Тому це треба міняти...

# Рефакторинг

## Математичні операції

Рефакторинг Я вирішив почати зі створення класу для математичних операцій. Його архітектура вийшла доволі простою: просто оголошення функції, її параметри, певна операція всередині та значення, яке повертає.

*Приклад однієї з функцій:*

```
fun add(firstNumber: Double, secondNumber: Double): Double  
{  
    return firstNumber + secondNumber  
}
```

## Великі Switch

Далі Я вирішив прибрати великі Switch. Ідея вирішення прийшло доволі швидко: на одній з лекцій була тема, яка чудово підходить для цього, а саме заміна умовних операторів поліморфізмом. Коли Я вперше почув про це, то Мені була незрозуміла її реалізація, тому Я не сильно приділяв увагу цьому способу, але спробувати хотів. І фінальний результат Мене вразив. Усе працювало, а код став набагато краще! І тема, яка нещодавно була неясною, відразу почала мати вагому роль у рефакторингу Мого коду.

*Приклад коду до рефакторингу:*

```
...  
multiply_btn.setOnClickListener { calculate("**") }  
...  
private fun calculate(operationChar: Char) {  
    when (operation) {  
        '+' -> {  
            result = MathClass.Add(tempValue1.toDouble(), tempValue2.toDouble())  
        }  
        '-' -> {  
            result = MathClass.Subtract(tempValue1.toDouble(), tempValue2.toDouble())  
        }  
        '/' -> {
```

```

when {
    firstNumber.text=="0.0" && !secondNumber.text.isNullOrEmpty() -> {
        result = tempValue2.toDouble()
    }
    !firstNumber.text.isNullOrEmpty() && !secondNumber.text.isNullOrEmpty() ->
    {
        result = MathClass.Divide(tempValue1.toDouble(), tempValue2.toDouble())
    }
}
}
'*' -> {
    when {
        firstNumber.text=="0.0" && !secondNumber.text.isNullOrEmpty() -> {
            result = tempValue2.toDouble()
        }
        !firstNumber.text.isNullOrEmpty() && !secondNumber.text.isNullOrEmpty() ->
        {
            result = MathClass.Multiply(tempValue1.toDouble(), tempValue2.toDouble())
        }
    }
}
}
}

```

Тобто код вище отримувач повідомлення від кнопки з певним символом та в функції `calculate()` у залежності від параметру у великому `Switch` робив перевірку.

Мало того, що це виглядає неестетично, так ще й читаємість та масштабування стають набагато гіршими.

А от після переходу на поліморфізм усе стало набагато краще:

*Приклад коду після рефакторингу:*

...

```
multiply_btn.setOnClickListener { calculate('*', OperatorMultiply(), false) }
```

...

```
open class Operator {
```

```
    open fun checkTwoNumbers(firstNumber: Double, secondNumber: Double): Double
```

```

{
    var mathClass = MathOperations()

    var result: Double = 0.0

    return result
}

open fun checkOneNumber(firstNumber: Double): Double
{
    var mathClass = MathOperations()

    var result: Double = 0.0

    return result
}
}

...

class OperatorMultiply : Operator(){
    override fun checkTwoNumbers(firstNumber: Double, secondNumber: Double): Double {
        var mathClass = MathOperations()

        return mathClass.multiply(firstNumber, secondNumber)
    }
}

...

private fun calculate(operationChar: Char, OperatorClass: Operator, oneNumberOperation: Boolean)
{
    ...

    when {
        !firstNumber.text.isNullOrEmpty() && !secondNumber.text.isNullOrEmpty() ->
        {
            result = OperatorClass.checkTwoNumbers(tempValue1.toDouble(),
tempValue2.toDouble())
        }
    }

    ...

    !firstNumber.text.isNullOrEmpty() && oneNumberOperation ->{
        result = OperatorClass.checkOneNumber(tempValue1.toDouble())
    }
}
}

```

```
...  
    }  
}
```

Код вище працює так:

Як один з параметрів відправляється екземпляр похідного класу, який має функції, прописані чітко для тої чи іншої операції, а за допомогою перевірок програма обирає, яку саме функцію класу викликати.

### Інші режими калькулятора

Однією з вимог є робота калькулятора в трьох режимах: 10-му, 2-му та 16-му.

Робити окремі класи та функції для обрахування чисел в різних формах у Мене не було бажання, адже, на Мою думку, це було зайвим. Тому Я використав хитрість: при проведенні операцій числа переводяться у 10-ву систему, приходять у функцію, яка була закладена класом `Operator`, повертається результат, який потім перетворюється у потрібну систему числення.

Для бінарного калькулятора Я використав функції `convertBinaryToDecimal()` та `convertDecimalToBinary()`, які знаходяться у класі `MathOperations`, а для 16-го – вбудовані функції для переводу: `Integer.parseInt(tempValue1,16)` та `Integer.toHexString(tempResult)`



# Вихідний текст програми

activity\_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/background_dark"
    android:orientation="vertical"
    >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="70sp"
        >

        <ImageView
            android:id="@+id/options_btn"
            style="@style/NumberButtonStyle"
            android:layout_width="120dp"
            android:layout_height="match_parent"
            android:ellipsize="start"
            android:gravity="end"
            android:scaleType="center"
            android:singleLine="true"
            android:src="@drawable/baseline_settings_white_18dp"
            android:textColor="@color/numberButton" />

        <HorizontalScrollView
            android:layout_width="292dp"
            android:layout_height="fill_parent"
            android:layout_marginRight="10dp"
            android:scrollbars="horizontal">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_gravity="right"
                android:ellipsize="end"
                android:gravity="end"
                android:orientation="horizontal"
                android:textAlignment="textEnd">

                <TextView
                    android:id="@+id/firstNumber"
                    android:layout_width="wrap_content"
                    android:layout_height="70sp"
                    android:gravity="end"
                    android:scrollbars="horizontal"
                    android:scrollHorizontally="true"
                    android:singleLine="true"
                    android:textColor="@color/gray"
                    android:textSize="40sp" />

            </LinearLayout>

        </HorizontalScrollView>
    </LinearLayout>
</HorizontalScrollView>
```

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"  
android:layout\_height="90sp">

<TextView

android:id="@+id/procedure"  
android:layout\_width="59dp"  
android:layout\_height="match\_parent"  
android:ellipsize="start"  
android:gravity="center"  
android:singleLine="true"  
android:textColor="@color/gray"  
android:textSize="30sp" />

<HorizontalScrollView

android:id="@+id/second\_scroll"  
android:layout\_width="302dp"  
android:layout\_height="fill\_parent"  
android:scrollbars="horizontal">

<LinearLayout

android:layout\_width="wrap\_content"  
android:layout\_height="match\_parent"  
android:layout\_gravity="right"  
android:orientation="horizontal">

<TextView

android:id="@+id/secondNumber"  
android:layout\_width="wrap\_content"  
android:layout\_height="match\_parent"  
android:scrollbars="horizontal"  
android:scrollHorizontally="true"  
android:singleLine="true"  
android:textColor="@color/white"  
android:textSize="50sp" />

</LinearLayout>

</HorizontalScrollView>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"  
android:layout\_height="match\_parent"  
android:orientation="vertical">

<LinearLayout

android:layout\_width="match\_parent"  
android:layout\_height="0dp"  
android:layout\_weight="1"  
android:orientation="horizontal">

<TextView

android:id="@+id/clear\_btn"  
style="@style/OperationButtonStyle"  
android:text="C"  
android:textSize="30sp" />

<TextView

android:id="@+id/plus\_minus\_btn"  
style="@style/OperationButtonStyle"

```
        android:text="±"
        android:textSize="40sp"/>
<TextView
    android:id="@+id/one_divide_x_btn"
    style="@style/OperationButtonStyle"
    android:text="1/X"
    android:textSize="30sp"/>
<TextView
    android:id="@+id/divide_btn"
    style="@style/OperationButtonStyle"
    android:text="/"
    android:textSize="40sp"/>
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/mod_btn"
        style="@style/OperationButtonStyle"
        android:text="%"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/degree_btn"
        style="@style/OperationButtonStyle"
        android:text="^"
        android:textSize="40sp"/>
    <TextView
        android:id="@+id/sin_btn"
        style="@style/OperationButtonStyle"
        android:text="sin"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/cos_btn"
        style="@style/OperationButtonStyle"
        android:text="cos"
        android:textSize="30sp"/>
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/seven_btn"
        style="@style/NumberButtonStyle"
        android:text="7"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/eight_btn"
        style="@style/NumberButtonStyle"
        android:text="8"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/nine_btn"
        style="@style/NumberButtonStyle"
        android:text="9"
```

```
        android:textSize="30sp"/>
<TextView
    android:id="@+id/multiply_btn"
    style="@style/OperationButtonStyle"
    android:text="*"
    android:textSize="40sp"/>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/four_btn"
        style="@style/NumberButtonStyle"
        android:text="4"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/five_btn"
        style="@style/NumberButtonStyle"
        android:text="5"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/six_btn"
        style="@style/NumberButtonStyle"
        android:text="6"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/minus_btn"
        style="@style/OperationButtonStyle"
        android:text="-"
        android:textSize="60sp"/>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/one_btn"
        style="@style/NumberButtonStyle"
        android:text="1"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/two_btn"
        style="@style/NumberButtonStyle"
        android:text="2"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/three_btn"
        style="@style/NumberButtonStyle"
        android:text="3"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/plus_btn"
        style="@style/OperationButtonStyle"
        android:text="+"
        android:textSize="40sp"/>
```

```
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/dot_btn"
        style="@style/NumberButtonStyle"
        android:text="."
        android:textSize="50sp"/>
    <TextView
        android:id="@+id/zero_btn"
        style="@style/NumberButtonStyle"
        android:text="0"
        android:textSize="30sp"/>
    <ImageView
        android:id="@+id/back_btn"
        style="@style/NumberButtonStyle"
        android:text="Back"
        android:textSize="30sp"
        android:src="@drawable/baseline_backspace_white_18dp"
        android:scaleType="center"/>
    <TextView
        android:id="@+id/equals_btn"
        style="@style/OperationButtonStyle"
        android:text="="
        android:textSize="50sp"/>
</LinearLayout>
</LinearLayout>
</LinearLayout>
```

## activity\_binary.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/background_dark"
    android:orientation="vertical"
    >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="70sp"
        >

        <ImageView
            android:id="@+id/options_btn"
            style="@style/NumberButtonStyle"
            android:layout_width="120dp"
            android:layout_height="match_parent"
            android:ellipsize="start"
            android:gravity="end"
            android:scaleType="center"
            android:singleLine="true"
            android:src="@drawable/baseline_settings_white_18dp"
            android:textColor="@color/numberButton" />

        <HorizontalScrollView
            android:layout_width="292dp"
            android:layout_height="fill_parent"
            android:layout_marginRight="10dp"
            android:scrollbars="horizontal">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_gravity="right"
                android:ellipsize="end"
                android:gravity="end"
                android:orientation="horizontal"
                android:textAlignment="textEnd">

                <TextView
                    android:id="@+id/firstNumber"
                    android:layout_width="wrap_content"
                    android:layout_height="70sp"
                    android:gravity="end"
                    android:scrollbars="horizontal"
                    android:scrollHorizontally="true"
                    android:singleLine="true"
                    android:textColor="@color/gray"
                    android:textSize="40sp" />

            </LinearLayout>

        </HorizontalScrollView>
    </LinearLayout>
</HorizontalScrollView>
```

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"  
android:layout\_height="90sp">

<TextView

android:id="@+id/procedure"  
android:layout\_width="59dp"  
android:layout\_height="match\_parent"  
android:ellipsize="start"  
android:gravity="center"  
android:singleLine="true"  
android:textColor="@color/gray"  
android:textSize="30sp" />

<HorizontalScrollView

android:id="@+id/second\_scroll"  
android:layout\_width="302dp"  
android:layout\_height="fill\_parent"  
android:scrollbars="horizontal">

<LinearLayout

android:layout\_width="wrap\_content"  
android:layout\_height="match\_parent"  
android:layout\_gravity="right"  
android:orientation="horizontal">

<TextView

android:id="@+id/secondNumber"  
android:layout\_width="wrap\_content"  
android:layout\_height="match\_parent"  
android:scrollbars="horizontal"  
android:scrollHorizontally="true"  
android:singleLine="true"  
android:textColor="@color/white"  
android:textSize="50sp" />

</LinearLayout>

</HorizontalScrollView>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"  
android:layout\_height="match\_parent"  
android:orientation="vertical">

<LinearLayout

android:layout\_width="match\_parent"  
android:layout\_height="0dp"  
android:layout\_weight="1"  
android:orientation="horizontal">

<TextView

android:id="@+id/clear\_btn"  
style="@style/OperationButtonStyle"  
android:text="C"  
android:textSize="60sp" />

<TextView

android:id="@+id/plus\_minus\_btn"  
style="@style/OperationButtonStyle"

```

        android:text="±"
        android:textSize="60sp"/>
<TextView
    android:id="@+id/degree_btn"
    style="@style/OperationButtonStyle"
    android:text="^"
    android:textSize="60sp"/>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/divide_btn"
        style="@style/OperationButtonStyle"
        android:text="/"
        android:textSize="60sp"/>
    <TextView
        android:id="@+id/mod_btn"
        style="@style/OperationButtonStyle"
        android:text="%"
        android:textSize="60sp"/>
    <TextView
        android:id="@+id/sin_btn"
        style="@style/OperationButtonStyle"
        android:text="sin"
        android:textSize="40sp"/>
    <TextView
        android:id="@+id/cos_btn"
        style="@style/OperationButtonStyle"
        android:text="cos"
        android:textSize="40sp"/>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/minus_btn"
        style="@style/OperationButtonStyle"
        android:text="-"
        android:textSize="60sp"/>
    <TextView
        android:id="@+id/plus_btn"
        style="@style/OperationButtonStyle"
        android:text="+"
        android:textSize="60sp"/>
    <TextView
        android:id="@+id/multiply_btn"
        style="@style/OperationButtonStyle"
        android:text="*"
        android:textSize="60sp"/>

```



```
<TextView
    android:id="@+id/equals_btn"
    style="@style/OperationButtonStyle"
    android:text="="
    android:textSize="60sp"/>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/one_btn"
        style="@style/NumberButtonStyle"
        android:text="1"
        android:textSize="60sp"/>
    <TextView
        android:id="@+id/zero_btn"
        style="@style/NumberButtonStyle"
        android:text="0"
        android:textSize="60sp"/>
    <ImageView
        android:id="@+id/back_btn"
        style="@style/NumberButtonStyle"
        android:text="Back"
        android:textSize="60sp"
        android:src="@drawable/baseline_backspace_white_18dp"
        android:scaleType="center"/>
</LinearLayout>
</LinearLayout>
</LinearLayout>
```

## activity\_hexadecimal.xml:

```
<?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:background="@android:color/background_dark"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="70sp"
        >

        <ImageView
            android:id="@+id/options_btn"
            style="@style/NumberButtonStyle"
            android:layout_width="120dp"
            android:layout_height="match_parent"
            android:ellipsize="start"
            android:gravity="end"
            android:scaleType="center"
            android:singleLine="true"
            android:src="@drawable/baseline_settings_white_18dp"
            android:textColor="@color/numberButton" />

        <HorizontalScrollView
            android:layout_width="292dp"
            android:layout_height="fill_parent"
            android:layout_marginRight="10dp"
            android:scrollbars="horizontal">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_gravity="right"
                android:ellipsize="end"
                android:gravity="end"
                android:orientation="horizontal"
                android:textAlignment="textEnd">

                <TextView
                    android:id="@+id/firstNumber"
                    android:layout_width="wrap_content"
                    android:layout_height="70sp"
                    android:gravity="end"
                    android:scrollbars="horizontal"
                    android:scrollHorizontally="true"
                    android:singleLine="true"
                    android:textColor="@color/gray"
                    android:textSize="40sp" />

            </LinearLayout>

        </HorizontalScrollView>

    </LinearLayout>

</LinearLayout>
```

```
android:layout_width="match_parent"
android:layout_height="80sp">
```

```
<TextView
    android:id="@+id/procedure"
    android:layout_width="59dp"
    android:layout_height="match_parent"
    android:ellipsize="start"
    android:gravity="center"
    android:singleLine="true"
    android:textColor="@color/gray"
    android:textSize="30sp" />
```

```
<HorizontalScrollView
    android:id="@+id/second_scroll"
    android:layout_width="302dp"
    android:layout_height="fill_parent"
    android:scrollbars="horizontal">
```

```
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="right"
    android:orientation="horizontal">
```

```
<TextView
    android:id="@+id/secondNumber"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:scrollbars="horizontal"
    android:scrollHorizontally="true"
    android:singleLine="true"
    android:textColor="@color/white"
    android:textSize="50sp" />
```

```
</LinearLayout>
```

```
</HorizontalScrollView>
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="horizontal">
```

```
<TextView
    android:id="@+id/clear_btn"
    style="@style/OperationButtonStyle"
    android:text="C"
    android:textSize="30sp" />
```

```
<TextView
    android:id="@+id/plus_minus_btn"
    style="@style/OperationButtonStyle"
    android:text="±"
    android:textSize="30sp" />
```

```
<TextView
    android:id="@+id/mod_btn"
```

```
style="@style/OperationButtonStyle"
android:text="%"
android:textSize="30sp"/>
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/d_btn"
        style="@style/NumberButtonStyle"
        android:text="D"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/e_btn"
        style="@style/NumberButtonStyle"
        android:text="E"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/f_btn"
        style="@style/NumberButtonStyle"
        android:text="F"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/degree_btn"
        style="@style/OperationButtonStyle"
        android:text="^"
        android:textSize="40sp"/>
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/a_btn"
        style="@style/NumberButtonStyle"
        android:text="A"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/b_btn"
        style="@style/NumberButtonStyle"
        android:text="B"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/c_btn"
        style="@style/NumberButtonStyle"
        android:text="C"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/divide_btn"
        style="@style/OperationButtonStyle"
        android:text="/"
        android:textSize="40sp"/>
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/seven_btn"
        style="@style/NumberButtonStyle"
        android:text="7"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/eight_btn"
        style="@style/NumberButtonStyle"
        android:text="8"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/nine_btn"
        style="@style/NumberButtonStyle"
        android:text="9"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/multiply_btn"
        style="@style/OperationButtonStyle"
        android:text="*"
        android:textSize="40sp"/>
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/four_btn"
        style="@style/NumberButtonStyle"
        android:text="4"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/five_btn"
        style="@style/NumberButtonStyle"
        android:text="5"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/six_btn"
        style="@style/NumberButtonStyle"
        android:text="6"
        android:textSize="30sp"/>
    <TextView
        android:id="@+id/minus_btn"
        style="@style/OperationButtonStyle"
        android:text="."
        android:textSize="60sp"/>
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="horizontal">
```

```
<TextView
    android:id="@+id/one_btn"
    style="@style/NumberButtonStyle"
    android:text="1"
    android:textSize="30sp"/>
<TextView
    android:id="@+id/two_btn"
    style="@style/NumberButtonStyle"
    android:text="2"
    android:textSize="30sp"/>
<TextView
    android:id="@+id/three_btn"
    style="@style/NumberButtonStyle"
    android:text="3"
    android:textSize="30sp"/>
<TextView
    android:id="@+id/plus_btn"
    style="@style/OperationButtonStyle"
    android:text="+"
    android:textSize="40sp"/>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/zero_btn"
        style="@style/NumberButtonStyle"
        android:text="0"
        android:textSize="30sp"/>
    <ImageView
        android:id="@+id/back_btn"
        style="@style/NumberButtonStyle"
        android:text="Back"
        android:textSize="30sp"
        android:src="@drawable/baseline_backspace_white_18dp"
        android:scaleType="center"/>
    <TextView
        android:id="@+id/equals_btn"
        style="@style/OperationButtonStyle"
        android:text="="
        android:textSize="50sp"/>
</LinearLayout>
</LinearLayout>
</LinearLayout>
```

## MainActivity.kt:

```
package com.example.calculator_oop_kotlin_project

import android.content.Intent
import android.os.Bundle
import android.widget.HorizontalScrollView
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity()
{
    var operation: Char = ' '

    lateinit var lastOperator: Operator

    override fun onCreate(savedInstanceState: Bundle?)
    {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        firstNumber.text="0.0"
        setOnClickListener()

        // set listener for every button
        private fun setOnClickListener()
        {
            options_btn.setOnClickListener { options() }
            clear_btn.setOnClickListener { clearAllText() }
            plus_minus_btn.setOnClickListener { calculate('±', OperatorPlusMinus(), true) }
            one_divide_x_btn.setOnClickListener { calculate('R', OperatorReciprocal(), true) }
            sin_btn.setOnClickListener { calculate('S', OperatorSin(), true) }
            cos_btn.setOnClickListener { calculate('C', OperatorCos(), true) }
            divide_btn.setOnClickListener { calculate('/', OperatorDivide(), false) }
            multiply_btn.setOnClickListener { calculate('*', OperatorMultiply(), false) }
            minus_btn.setOnClickListener { calculate('-', OperatorMinus(), false) }
            plus_btn.setOnClickListener { calculate('+', OperatorPlus(), false) }
            equals_btn.setOnClickListener { calculate(operation, lastOperator, false) }
            back_btn.setOnClickListener { clearLastCharacter() }
            mod_btn.setOnClickListener { calculate('%', OperatorMod(), false) }
            degree_btn.setOnClickListener { calculate('^', OperatorDegree(), false) }

            nine_btn.setOnClickListener { appendText("9") }
            eight_btn.setOnClickListener { appendText("8") }
            seven_btn.setOnClickListener { appendText("7") }
            six_btn.setOnClickListener { appendText("6") }
            five_btn.setOnClickListener { appendText("5") }
            four_btn.setOnClickListener { appendText("4") }
            three_btn.setOnClickListener { appendText("3") }
            two_btn.setOnClickListener { appendText("2") }
            one_btn.setOnClickListener { appendText("1") }
            zero_btn.setOnClickListener { appendText("0") }
            dot_btn.setOnClickListener { appendText(".") }
        }

        // add text to second TextView and automatically scrolls to the end
        private fun appendText(number: String)
        {

```

```

secondNumber.append(number);
second_scroll.post { second_scroll.fullScroll(HorizontalScrollView.FOCUS_RIGHT) }
}

// clear all text
private fun clearAllText()
{
    firstNumber.text="0.0"
    secondNumber.text=""
    operation=' '
    procedure.text=""
}

// clear last character
private fun clearLastCharacter()
{
    val tempString = secondNumber.text.toString()
    if (tempString.isNotEmpty())
    {
        secondNumber.text = tempString.substring(0, tempString.length - 1)
    }
}

// main function for calculating
private fun calculate(operationChar: Char, OperatorClass: Operator, oneNumberOperation: Boolean) {
    // current operation
    operation = operationChar

    lastOperator = OperatorClass

    var tempValue1 = firstNumber.text.toString()
    var tempValue2 = secondNumber.text.toString()

    var result = tempValue1.toDouble()

    when {
        (firstNumber.text=="0.0" || firstNumber.text=="-0.0" ||
            firstNumber.text=="Infinity" || firstNumber.text=="-Infinity")
            && !secondNumber.text.isNullOrEmpty() -> {
            result = tempValue2.toDouble()
        }

        !firstNumber.text.isNullOrEmpty() && !secondNumber.text.isNullOrEmpty() ->
        {
            result = OperatorClass.checkTwoNumbers(tempValue1.toDouble(), tempValue2.toDouble())
        }

        !firstNumber.text.isNullOrEmpty() && oneNumberOperation ->{
            result = OperatorClass.checkOneNumber(tempValue1.toDouble())
        }

        firstNumber.text.isNullOrEmpty() && !secondNumber.text.isNullOrEmpty() -> {
            firstNumber.text = secondNumber.text
            firstNumber.append(".0")
            secondNumber.text = ""
        }
    }

    // do this after calling calculate
    firstNumber.text = result.toString()
}

```



```

secondNumber.text = ""
procedure.text=operation.toString()
}

// options button
private fun options()
{
    var tempString = ""
    val listItems = arrayOf("Decimal", "Binary", "Hexadecimal")
    val mBuilder = AlertDialog.Builder(this@MainActivity)
    mBuilder.setTitle("Type of Calculator")
    mBuilder.setSingleChoiceItems(listItems, -1) { dialogInterface, i ->
        tempString = listItems[i]
        dialogInterface.dismiss()

        when (tempString) {
            "Decimal" -> {
                startActivity(Intent(this, MainActivity::class.java))
            }
            "Binary" -> {
                startActivity(Intent(this, BinaryActivity::class.java))
            }
            "Hexadecimal" -> {
                startActivity(Intent(this, HexadecimalActivity::class.java))
            }
        }
    }
}

// Set the neutral/cancel button click listener
mBuilder.setNeutralButton("Cancel") { dialog, which ->
    // Do something when click the neutral button
    dialog.cancel()
}

val mDialog = mBuilder.create()
mDialog.show()
}
}

```

## BinaryActivity.kt:

```
package com.example.calculator_oop_kotlin_project
```

```
import android.content.Intent
import android.os.Bundle
import android.widget.HorizontalScrollView
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import kotlinx.android.synthetic.main.activity_main.*
```

```
class BinaryActivity : AppCompatActivity() {
    var operation: Char = ' '
    var mathClass = MathOperations()
    lateinit var lastOperator: Operator

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_binary)
        firstNumber.text="0"
        setOnClickListeners()
    }

    // set listener for every button
    private fun setOnClickListeners()
    {
        options_btn.setOnClickListener { options() }
        clear_btn.setOnClickListener { clearAllText() }
        plus_minus_btn.setOnClickListener { calculate('±', OperatorPlusMinus(), true) }
        sin_btn.setOnClickListener { calculate('S', OperatorSin(), true) }
        cos_btn.setOnClickListener { calculate('C', OperatorCos(), true) }
        divide_btn.setOnClickListener { calculate('/', OperatorDivide(), false) }
        multiply_btn.setOnClickListener { calculate('*', OperatorMultiply(), false) }
        minus_btn.setOnClickListener { calculate('-', OperatorMinus(), false) }
        plus_btn.setOnClickListener { calculate('+', OperatorPlus(), false) }
        equals_btn.setOnClickListener { calculate(operation, lastOperator, false) }
        back_btn.setOnClickListener { clearLastCharacter() }
        mod_btn.setOnClickListener { calculate('%', OperatorMod(), false) }
        degree_btn.setOnClickListener { calculate('^', OperatorDegree(), false) }

        one_btn.setOnClickListener { appendText("1") }
        zero_btn.setOnClickListener { appendText("0") }
    }

    // add text to second TextView and automatically scrolls to the end
    private fun appendText(number: String)
    {
        secondNumber.append(number);
        second_scroll.post { second_scroll.fullScroll(HorizontalScrollView.FOCUS_RIGHT) }
    }

    // clear all text
    private fun clearAllText()
    {
        firstNumber.text="0"
        secondNumber.text=""
        operation=' '
        procedure.text=""
    }
}
```

```

// clear last character
private fun clearLastCharacter()
{
    val tempString = secondNumber.text.toString()
    if (tempString.isNotEmpty())
    {
        secondNumber.text = tempString.substring(0, tempString.length - 1)
    }
}

// main function for calculating
private fun calculate(operationChar: Char, OperatorClass: Operator, oneNumberOperation: Boolean) {
    // current operation
    operation = operationChar

    lastOperator = OperatorClass

    var tempResult: Int = 0

    var tempValue1 = firstNumber.text.toString()
    var tempValue2 = secondNumber.text.toString()

    var result = tempValue1.toInt()

    when {
        (firstNumber.text == "0" || firstNumber.text == "-0" ||
            firstNumber.text == "Infinity" || firstNumber.text == "-Infinity" )
            && !secondNumber.text.isNullOrEmpty() -> {
            result = tempValue2.toInt()
        }

        !firstNumber.text.isNullOrEmpty() && !secondNumber.text.isNullOrEmpty() ->
        {
            // convert to decimal
            var tempDecimalValue1 = mathClass.convertBinaryToDecimal(tempValue1.toFloat())
            var tempDecimalValue2 = mathClass.convertBinaryToDecimal(tempValue2.toFloat())

            tempResult =
            OperatorClass.checkTwoNumbers(tempDecimalValue1.toDouble(), tempDecimalValue2.toDouble()).toInt()

            result = mathClass.convertDecimalToBinary(tempResult.toFloat())
        }

        !firstNumber.text.isNullOrEmpty() && oneNumberOperation -> {
            result = OperatorClass.checkOneNumber(tempValue1.toDouble()).toInt()
        }

        firstNumber.text.isNullOrEmpty() && !secondNumber.text.isNullOrEmpty() -> {
            firstNumber.text = secondNumber.text
            secondNumber.text = ""
        }
    }

    // do this after calling calculate
    firstNumber.text = result.toString()
    secondNumber.text = ""
    procedure.text = operation.toString()
}

// options button

```

```

private fun options()
{
    var tempString = ""
    val listItems = arrayOf("Decimal", "Binary", "Hexadecimal")
    val mBuilder = AlertDialog.Builder(this@BinaryActivity)
    mBuilder.setTitle("Type of Calculator")
    mBuilder.setSingleChoiceItems(listItems, -1) { dialogInterface, i ->
        tempString = listItems[i]
        dialogInterface.dismiss()

        when (tempString) {
            "Decimal" -> {
                startActivity(Intent(this, MainActivity::class.java))
            }
            "Binary" -> {
                startActivity(Intent(this, BinaryActivity::class.java))
            }
            "Hexadecimal" -> {
                startActivity(Intent(this, HexadecimalActivity::class.java))
            }
        }
    }

    // Set the neutral/cancel button click listener
    mBuilder.setNeutralButton("Cancel") { dialog, which ->
        // Do something when click the neutral button
        dialog.cancel()
    }

    val mDialog = mBuilder.create()
    mDialog.show()
}
}

```

## HexadecimalActivity.kt:

```
package com.example.calculator_oop_kotlin_project
```

```
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.HorizontalScrollView
import androidx.appcompat.app.AlertDialog
import kotlinx.android.synthetic.main.activity_hexadecimal.*
```

```
class HexadecimalActivity : AppCompatActivity() {
    var operation: Char = ''
    lateinit var lastOperator: Operator
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_hexadecimal)
        firstNumber.text="0"
        setOnClickListeners()
    }
```

```
// set listener for every button
```

```
private fun setOnClickListeners()
{
    options_btn.setOnClickListener { options() }
    clear_btn.setOnClickListener { clearAllText() }
    plus_minus_btn.setOnClickListener { calculate('±', OperatorPlusMinus(), true) }
    divide_btn.setOnClickListener { calculate('/', OperatorDivide(), false) }
    multiply_btn.setOnClickListener { calculate('*', OperatorMultiply(), false) }
    minus_btn.setOnClickListener { calculate('-', OperatorMinus(), false) }
    plus_btn.setOnClickListener { calculate('+', OperatorPlus(), false) }
    equals_btn.setOnClickListener { calculate(operation, lastOperator, false) }
    back_btn.setOnClickListener { clearLastCharacter() }
    mod_btn.setOnClickListener { calculate('%', OperatorMod(), false) }
    degree_btn.setOnClickListener { calculate('^', OperatorDegree(), false) }
```

```
    f_btn.setOnClickListener { appendText("F") }
    e_btn.setOnClickListener { appendText("E") }
    d_btn.setOnClickListener { appendText("D") }
    c_btn.setOnClickListener { appendText("C") }
    b_btn.setOnClickListener { appendText("B") }
    a_btn.setOnClickListener { appendText("A") }
    nine_btn.setOnClickListener { appendText("9") }
    eight_btn.setOnClickListener { appendText("8") }
    seven_btn.setOnClickListener { appendText("7") }
    six_btn.setOnClickListener { appendText("6") }
    five_btn.setOnClickListener { appendText("5") }
    four_btn.setOnClickListener { appendText("4") }
    three_btn.setOnClickListener { appendText("3") }
    two_btn.setOnClickListener { appendText("2") }
    one_btn.setOnClickListener { appendText("1") }
    zero_btn.setOnClickListener { appendText("0") }
}
```

```
// add text to second TextView and automatically scrolls to the end
```

```
private fun appendText(number: String)
{
    secondNumber.append(number);
    second_scroll.post { second_scroll.fullScroll(HorizontalScrollView.FOCUS_RIGHT) }
```

```

}

// clear all text
private fun clearAllText()
{
    firstNumber.text="0"
    secondNumber.text=""
    operation=" "
    procedure.text=""
}

// clear last character
private fun clearLastCharacter()
{
    val tempString = secondNumber.text.toString()
    if (tempString.isNotEmpty())
    {
        secondNumber.text = tempString.substring(0, tempString.length - 1)
    }
}

// main function for calculating
private fun calculate(operationChar: Char, OperatorClass: Operator, oneNumberOperation: Boolean) {
    // current operation
    operation = operationChar

    lastOperator = OperatorClass

    var tempResult:Int=0

    var tempValue1 = firstNumber.text.toString()
    var tempValue2 = secondNumber.text.toString()

    var result = tempValue1

    when {
        (firstNumber.text=="0" || firstNumber.text=="-0" ||
            firstNumber.text=="Infinity" || firstNumber.text=="-Infinity")
            && !secondNumber.text.isNullOrEmpty() -> {
            result = tempValue2
        }

        !firstNumber.text.isNullOrEmpty() && !secondNumber.text.isNullOrEmpty() ->
        {
            // convert to decimal
            var tempDecimalValue1 = Integer.parseInt(tempValue1,16)
            var tempDecimalValue2 = Integer.parseInt(tempValue2,16)

            tempResult = OperatorClass.checkTwoNumbers(tempDecimalValue1.toDouble(),
                tempDecimalValue2.toDouble()).toInt()

            result = Integer.toHexString(tempResult)
        }

        !firstNumber.text.isNullOrEmpty() && oneNumberOperation -> {
            result = OperatorClass.checkOneNumber(tempValue1.toDouble()).toInt().toString()
        }

        firstNumber.text.isNullOrEmpty() && !secondNumber.text.isNullOrEmpty() -> {
            firstNumber.text = secondNumber.text
        }
    }
}

```

```

        secondNumber.text = ""
    }
}

// do this after calling calculate
firstNumber.text = result.toUpperCase()
secondNumber.text = ""
procedure.text=operation.toString()
}

// options button
private fun options()
{
    var tempString = ""
    val listItems = arrayOf("Decimal", "Binary", "Hexadecimal")
    val mBuilder = AlertDialog.Builder(this@HexadecimalActivity)
    mBuilder.setTitle("Type of Calculator")
    mBuilder.setSingleChoiceItems(listItems, -1) { dialogInterface, i ->
        tempString = listItems[i]
        dialogInterface.dismiss()

        when (tempString) {
            "Decimal" -> {
                startActivity(Intent(this, MainActivity::class.java))
            }
            "Binary" -> {
                startActivity(Intent(this, BinaryActivity::class.java))
            }
            "Hexadecimal" -> {
                startActivity(Intent(this, HexadecimalActivity::class.java))
            }
        }
    }
}

// Set the neutral/cancel button click listener
mBuilder.setNeutralButton("Cancel") { dialog, which ->
    // Do something when click the neutral button
    dialog.cancel()
}

val mDialog = mBuilder.create()
mDialog.show()
}
}

```

# MathOperations.kt:

```
package com.example.calculator_oop_kotlin_project
```

```
import kotlin.math.pow
```

```
class MathOperations {  
    fun add(firstNumber: Double, secondNumber: Double): Double  
    {  
        return firstNumber + secondNumber  
    }  
  
    fun subtract(firstNumber: Double, secondNumber: Double): Double  
    {  
        return firstNumber - secondNumber  
    }  
  
    fun divide(firstNumber: Double, secondNumber: Double): Double  
    {  
        return firstNumber / secondNumber  
    }  
  
    fun multiply(firstNumber: Double, secondNumber: Double): Double  
    {  
        return firstNumber * secondNumber  
    }  
  
    fun mod(firstNumber: Double, secondNumber: Double): Double  
    {  
        return firstNumber % secondNumber  
    }  
  
    fun degree(firstNumber: Double, secondNumber: Double): Double  
    {  
        return firstNumber.pow(secondNumber)  
    }  
  
    fun plusMinus(firstNumber: Double): Double  
    {  
        return firstNumber * -1  
    }  
  
    fun reciprocal(firstNumber: Double): Double  
    {  
        return 1 / firstNumber  
    }  
  
    fun sin(firstNumber: Double): Double  
    {  
        return kotlin.math.sin(firstNumber)  
    }  
  
    fun cos(firstNumber: Double): Double  
    {  
        return kotlin.math.cos(firstNumber)  
    }  
  
    fun convertBinaryToDecimal(num: Float): Int {  
        var num = num  
        var decNum = 0  
    }
```



```

var i = 0
var residual: Float
while (num != 0f) {
    residual = num % 10
    num /= 10
    decNum += (residual * Math.pow(2.0, i.toDouble())).toInt()
    ++i
}
return decNum
}

fun convertDecimalToBinary(n: Float): Int {
    var n = n
    var binNum: Int = 0
    var residual: Int
    var i = 1
    while (n != 0f) {
        residual = (n % 2).toInt()
        n /= 2
        binNum += (residual * i)
        i *= 10
    }
    return binNum
}
}

```

## Operator.kt:

```
package com.example.calculator_oop_kotlin_project
```

```
open class Operator {
    open fun checkTwoNumbers(firstNumber: Double, secondNumber: Double): Double
    {
        var mathClass = MathOperations()

        var result: Double = 0.0
        return result
    }
    open fun checkOneNumber(firstNumber: Double): Double
    {
        var mathClass = MathOperations()

        var result: Double = 0.0
        return result
    }
}

class OperatorPlus : Operator(){
    override fun checkTwoNumbers(firstNumber: Double, secondNumber: Double): Double {
        var mathClass = MathOperations()

        return mathClass.add(firstNumber, secondNumber)
    }
}

class OperatorMinus : Operator(){
    override fun checkTwoNumbers(firstNumber: Double, secondNumber: Double): Double {
        var mathClass = MathOperations()

        return mathClass.subtract(firstNumber, secondNumber)
    }
}

class OperatorMod : Operator(){
    override fun checkTwoNumbers(firstNumber: Double, secondNumber: Double): Double {
        var mathClass = MathOperations()

        return mathClass.mod(firstNumber, secondNumber)
    }
}

class OperatorDegree : Operator(){
    override fun checkTwoNumbers(firstNumber: Double, secondNumber: Double): Double {
        var mathClass = MathOperations()

        return mathClass.degree(firstNumber, secondNumber)
    }
}

class OperatorMultiply : Operator(){
    override fun checkTwoNumbers(firstNumber: Double, secondNumber: Double): Double {
        var mathClass = MathOperations()

        return mathClass.multiply(firstNumber, secondNumber)
    }
}
```

```
class OperatorDivide : Operator0{
    override fun checkTwoNumbers(firstNumber: Double, secondNumber: Double): Double {
        var mathClass = MathOperations()

        return mathClass.divide(firstNumber, secondNumber)
    }
}
```

```
class OperatorSin : Operator0{
    override fun checkOneNumber(firstNumber: Double): Double {
        var mathClass = MathOperations()

        return mathClass.sin(firstNumber)
    }
}
```

```
class OperatorCos : Operator0{
    override fun checkOneNumber(firstNumber: Double): Double {
        var mathClass = MathOperations()

        return mathClass.cos(firstNumber)
    }
}
```

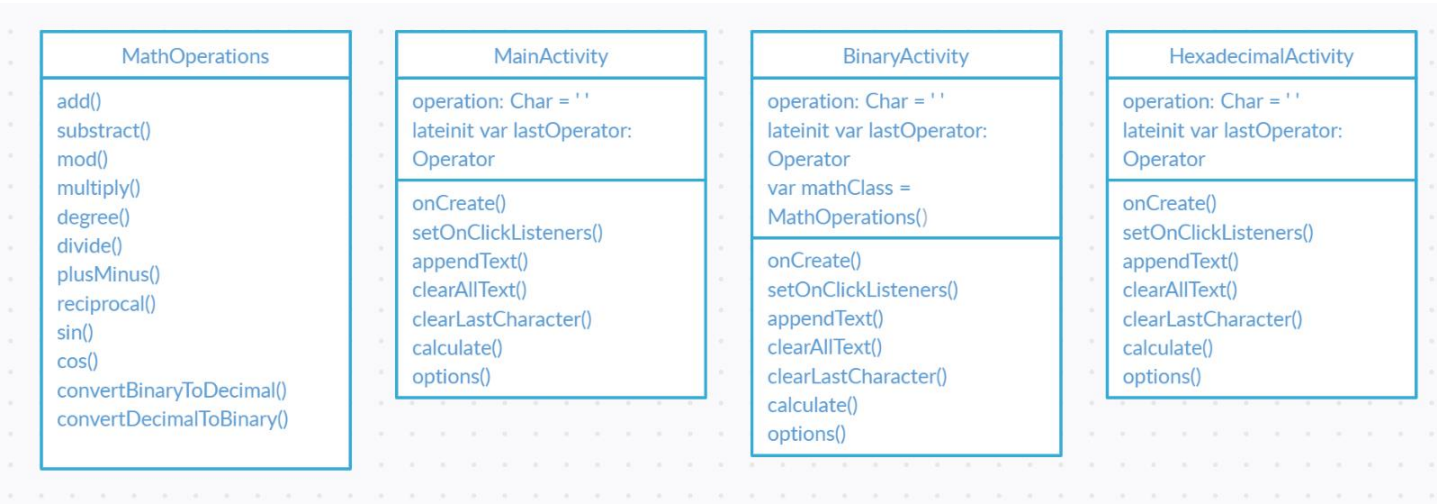
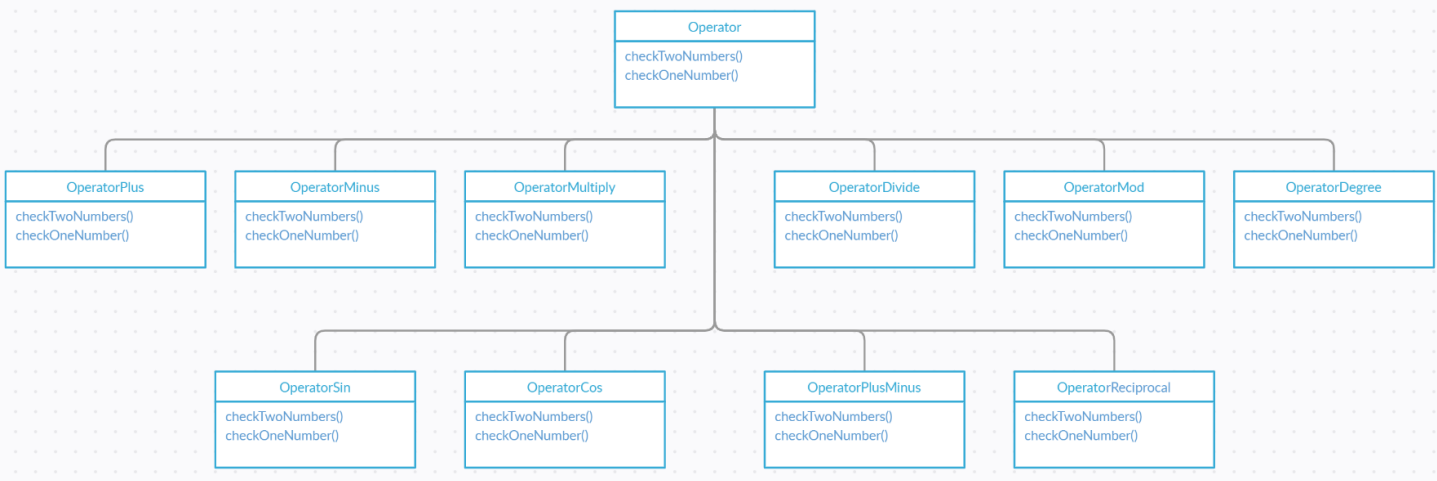
```
class OperatorReciprocal : Operator0{
    override fun checkOneNumber(firstNumber: Double): Double {
        var mathClass = MathOperations()

        return mathClass.reciprocal(firstNumber)
    }
}
```

```
class OperatorPlusMinus : Operator0{
    override fun checkOneNumber(firstNumber: Double): Double {
        var mathClass = MathOperations()

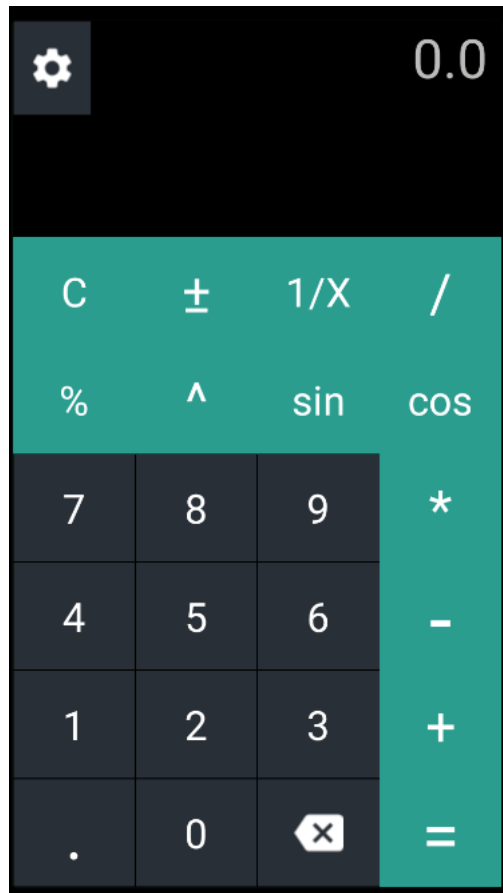
        return mathClass.plusMinus(firstNumber)
    }
}
```

# Діаграма класів:

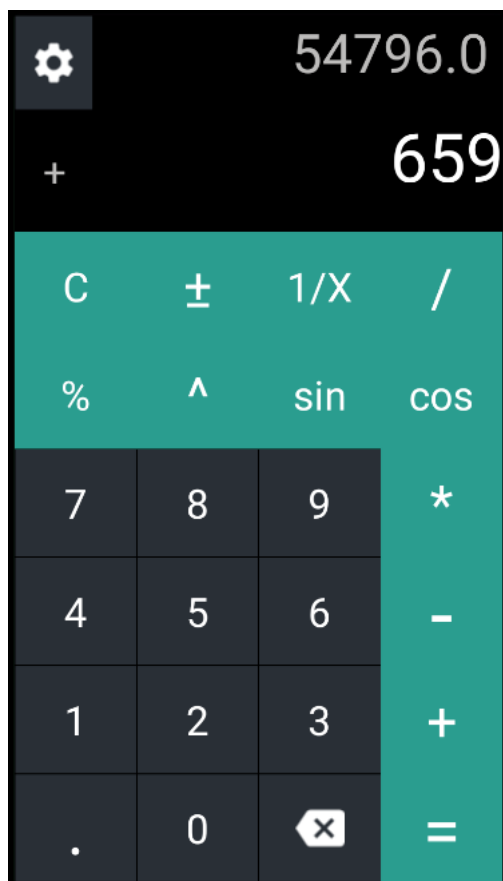


## Скріншоти:

Початкове вікно:



Введення цифр:



Меню переключення режимів:

Type of Calculator

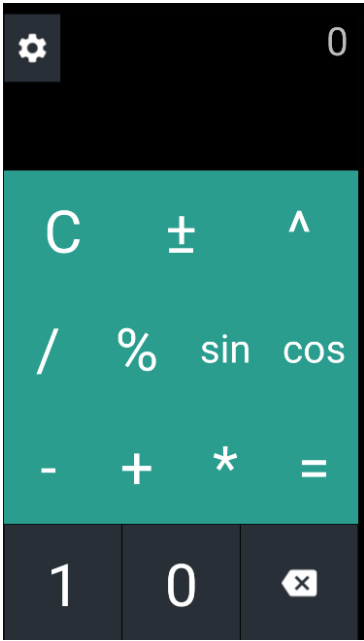
☐ Decimal

☐ Binary

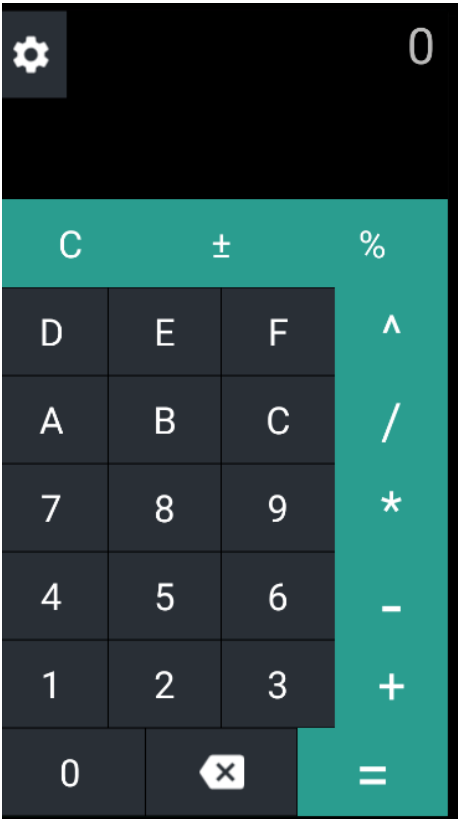
☐ Hexadecimal

CANCEL

Двійковий калькулятор:



Шістнадцятковий калькулятор:



## Висновок:

Мені було цікаво спробувати щось нове, як android + kotlin. Документація kotlin є доволі пізнавальною, а працювати з android studio було зручно. Під час роботи Я познайомився з Kotlin, Android, Android Studio, XML, поглибив знання ООП.