

5. 실습 보고서
- 계산기 실습 -

보 고 자 성 명	20181151 김 지 원
보 고 자 학 과	산 업 경 영 공 학 과
담 당 교 수 성 명	정 진 만 교 수 님
과 목 명	모 바 일 프 로 그 래밍 01분 반
제 출 날 짜	2020.04.19

- 목차 -

1. 서론	3p
1) 개요	
2) 요청사항 분석	
3) 추가사항	
4) 초기 계산기 디자인	
2. 본론	
1) activity_main.xml	4p
2) MainActivity.java	
① 소스 코드 원본	5p
② 소스 코드 분석	12p
3) 결과 화면	17p
3. 결론	
1) 나만의 계산기 장점	18p
2) 나만의 계산기 단점	18p
3) 배운 점	19p

1. 서론

1) 개요

Button과 TextView를 이용하여 자신만의 안드로이드용 계산기 앱을 제작한다. 흔히 사용하는 계산기와 동일한 방식으로 사칙연산을 Button으로 입력한 후 '='을 누르면 TextView 창에 결과 값이 도출된다.

2) 요청사항 분석

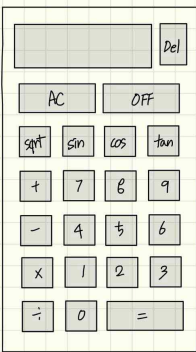
- ① clear : Button AC를 추가하여 TextView에 입력된 값을 초기화하여 '0'이 표시되게 한다.
- ② Backspace : Button Del을 추가하여 TextView에 입력된 값의 가장 낮은 자릿수의 숫자부터 한 자리씩 제거한다.
- ③ sqrt : 제곱근으로 Button sqrt를 추가하여 TextView의 입력된 값의 제곱근을 표시한다.
- ④ sin : Button sin을 추가하여 TextView의 입력된 값의 sin 값을 표시한다.
- ⑤ cos : Button cos을 추가하여 TextView의 입력된 값의 cos 값을 표시한다.
- ⑥ tan : Button tan를 추가하여 TextView의 입력된 값의 tan 값을 표시한다.

3) 추가사항

- OFF 기능

: 일반 계산기와 동일한 방식으로 OFF 기능을 추가하여 해당 Button 클릭 시, 앱이 종료된다.

4) 초기 계산기 디자인

	<ul style="list-style-type: none">- 결과 값이 출력되는 화면은 가장 상단에 위치시킨다.- Backspace는 컴퓨터 자판과 유사하게 우측 상단에 위치시켜 익숙한 조작을 할 수 있게 한다.- AC와 OFF는 상대적으로 사용 빈도가 낮아 Device를 사용 시 손이 잘 닿지 않는 상단에 배치한다.- 가장 많이 사용하는 숫자와 기본 사칙연산(+, -, x, /)을 손이 잘 닿는 하단에 배치한다.- '='은 가장 많이 사용하는 버튼으로 우측 하단에 사이즈를 확대하여 배치한다.
---	--

2. 본문

1) activity_main.xml

Component Tree

- LinearLayout (vertical)
 - LinearLayout (horizontal)
 - Ab Text_Rlt "0"
 - Bnt_Del "DEL"
 - LinearLayout (horizontal)
 - Bnt_AC "AC"
 - Bnt_OFF "OFF"
 - LinearLayout (horizontal)
 - Bnt_Sqrt "sqrt"
 - Bnt_Sin "sin"
 - Bnt_Cos "cos"
 - Bnt_Tan "tan"
 - LinearLayout (horizontal)
 - Bnt_Sum "+"
 - Bnt_7 "7"
 - Bnt_8 "8"
 - Bnt_9 "9"
 - LinearLayout (horizontal)
 - Bnt_Min "-"
 - Bnt_4 "4"
 - Bnt_5 "5"
 - Bnt_6 "6"
 - LinearLayout (horizontal)
 - Bnt_Mul "x"
 - Bnt_1 "1"
 - Bnt_2 "2"
 - Bnt_3 "3"
 - LinearLayout (horizontal)
 - Bnt_Div "/"
 - Bnt_0 "0"
 - Bnt_Eql "="

- vertical Linear Layout에 7개의 horizontal Linear Layout을 삽입한다.
- 각 Layout에 TextView와 Button을 삽입한다.
- 모든 위젯은 구분이 용이하도록 padding을 5dp 부여하여 Button 사이의 간격을 조정한다.
- 각 Button은 용도에 맞게 색으로 구별한다.
- 각 위젯의 weight를 조정하여 Button의 크기를 일정하게 디자인한다.

2) MainActivity.java

① 소스 코드 원본

```
package com.example.week5_cal;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    TextView outcome;

    float mArg;
    int state;
    float result;

    public void onBtnDel(View v)
    {
        String input = outcome.getText().toString();

        input = input.substring(0,input.length()-1);
        outcome.setText(input);

        if(input.equals(""))
        {
            outcome.setText("0");
        }

        Log.v("my.msg", "onBtnDel");
    }

    public void onBtnAC(View v)
    {
        outcome.setText("0");

        Log.v("my.msg", "onBtnAC");
    }

    public void onBtnOFF(View v)
    {
        finish();

        Log.v("my.msg", "onBtnOFF");
    }

    public void onBtnSqrt(View v)
    {
        String str = outcome.getText().toString();

        mArg = Float.valueOf(str);

        String check = Double.toString(Math.sqrt(mArg));
        if(check.length() > 15)
        {
            outcome.setText(String.format("%.12f",Math.sqrt(mArg)));
        }
        else if(check.length() <= 15)
        {
            outcome.setText(""+Math.sqrt(mArg));
        }

        if(Math.sqrt(mArg)-Math.round(Math.sqrt(mArg)) == 0)
        {

```

```

outcome.setText(""+Math.round(Math.sqrt(mArg)));
}

Log.v("my.msg", "onBtnSqrt");
}

public void onBtnSin(View v)
{
String str = outcome.getText().toString();

mArg = Float.valueOf(str);

String check = Double.toString(Math.sin(mArg));
if(check.length() > 15)
{
outcome.setText(String.format("%.12f", Math.sin(mArg)));
}
else if(check.length() <= 15)
{
outcome.setText(""+Math.sin(mArg));
}

if(Math.sin(mArg)-Math.round(Math.sqrt(Math.sin(mArg))) == 0)
{
outcome.setText(""+Math.round(Math.sin(mArg)));
}

Log.v("my.msg", "onBtnSin");
}

public void onBtnCos(View v)
{
String str = outcome.getText().toString();

mArg = Float.valueOf(str);

String check = Double.toString(Math.cos(mArg));
if(check.length() > 15)
{
outcome.setText(String.format("%.12f", Math.cos(mArg)));
}
else if(check.length() <= 15)
{
outcome.setText(""+Math.cos(mArg));
}

if(Math.cos(mArg)-Math.round(Math.sqrt(Math.cos(mArg))) == 0)
{
outcome.setText(""+Math.round(Math.cos(mArg)));
}

Log.v("my.msg", "onBtnCos");
}

public void onBtnTan(View v)
{
String str = outcome.getText().toString();

mArg = Float.valueOf(str);

String check = Double.toString(Math.tan(mArg));
if(check.length() > 15)
{
outcome.setText(String.format("%.12f", Math.tan(mArg)));
}
else if(check.length() <= 15)
{
outcome.setText(""+Math.tan(mArg));
}

if(Math.tan(mArg)-Math.round(Math.sqrt(Math.tan(mArg))) == 0)
{

```

```

outcome.setText(""+Math.round(Math.tan(mArg)));
}

Log.v("my.msg", "onBtnTan");
}

public void onBtnSum(View v)
{
String str = outcome.getText().toString();

//mArg = Integer.parseInt(str);
mArg = Float.valueOf(str);
outcome.setText("0");
state = 1;

Log.v("my.msg", "onBtnSum");
}

public void onBtnMin(View v)
{
String str = outcome.getText().toString();

//mArg = Integer.parseInt(str);
mArg = Float.valueOf(str);
outcome.setText("0");
state = 2;

Log.v("my.msg", "onBtnMin");
}

public void onBtnMul(View v)
{
String str = outcome.getText().toString();

//mArg = Integer.parseInt(str);
mArg = Float.valueOf(str);
outcome.setText("0");
state = 3;

Log.v("my.msg", "onBtnMul");
}

public void onBtnDiv(View v)
{
String str = outcome.getText().toString();

//mArg = Integer.parseInt(str);
mArg = Float.valueOf(str);
outcome.setText("0");
state = 4;

Log.v("my.msg", "onBtnDiv");
}

public void onBtnEq1(View v)
{
String str = outcome.getText().toString();
//int current = Integer.parseInt(str);
float current = Float.valueOf(str);

// + click
if(state == 1)
{
result = mArg + current;
}

// - click
if(state == 2)
{
result = mArg - current;
}
}

```

```

// x click
if(state == 3)
{
    result = mArg * current;
}

// / click
if(state == 4)
{
    result = mArg / current;
}

if(result-Math.round(result) == 0)
{
    outcome.setText(""+Math.round(result));
}

else if(result-Math.round(result) != 0)
{
    // Toast.makeText(getApplicationContext(),
    ""+Float.toString(result).length(), Toast.LENGTH_LONG).show();

    if(Float.toString(result).length() > 15)
    {
        outcome.setText(String.format("%.12",result));
    }
    else if(Float.toString(result).length() <= 15)
    {
        outcome.setText(""+result);
    }
    // outcome.setText(""+result);
}

Log.v("my.msg", "onBtnEq1");
}

public void onBtn0(View v)
{
    String input = outcome.getText().toString();

    if(input.equals("0"))
    {
        return;
    }

    else if(input.length() >=15 )
    {
        input = input;
    }

    else
    {
        input = input + "0";
    }

    outcome.setText(input);

    Log.v("my.msg", "onBtn0");
}

public void onBtn1(View v)
{
    String input = outcome.getText().toString();

    if(input.equals("0"))
    {
        input = "1";
    }

    else if(input.length() >=15 )

```



```

{
    input = input;
}

else
{
    input = input + "1";
}

outcome.setText(input);

Log.v("my.msg", "onBtn1");
}

public void onBtn2(View v)
{
    String input = outcome.getText().toString();

    if(input.equals("0"))
    {
        input = "2";
    }

    else if(input.length() >=15 )
    {
        input = input;
    }

    else
    {
        input = input + "2";
    }

    outcome.setText(input);

    Log.v("my.msg", "onBtn2");
}

public void onBtn3(View v)
{
    String input = outcome.getText().toString();

    if(input.equals("0"))
    {
        input = "3";
    }

    else if(input.length() >=15 )
    {
        input = input;
    }

    else
    {
        input = input + "3";
    }

    outcome.setText(input);

    Log.v("my.msg", "onBtn3");
}

public void onBtn4(View v)
{
    String input = outcome.getText().toString();

    if(input.equals("0"))
    {
        input = "4";
    }

    else if(input.length() >=15 )

```

```

{
    input = input;
}

else
{
    input = input + "4";
}

outcome.setText(input);

Log.v("my.msg", "onBtn4");
}

public void onBtn5(View v)
{
    String input = outcome.getText().toString();

    if(input.equals("0"))
    {
        input = "5";
    }

    else if(input.length() >=15 )
    {
        input = input;
    }

    else
    {
        input = input + "5";
    }

    outcome.setText(input);

    Log.v("my.msg", "onBtn5");
}

public void onBtn6(View v)
{
    String input = outcome.getText().toString();

    if(input.equals("0"))
    {
        input = "6";
    }

    else if(input.length() >=15 )
    {
        input = input;
    }

    else
    {
        input = input + "6";
    }

    outcome.setText(input);

    Log.v("my.msg", "onBtn6");
}

public void onBtn7(View v)
{
    String input = outcome.getText().toString();

    if(input.equals("0"))
    {
        input = "7";
    }

    else if(input.length() >=15 )

```

```

{
    input = input;
}

else
{
    input = input + "7";
}

outcome.setText(input);
Log.v("my.msg", "onBtn7");
}

public void onBtn8(View v)
{
    String input = outcome.getText().toString();

    if(input.equals("0"))
    {
        input = "8";
    }

    else if(input.length() >=15 )
    {
        input = input;
    }

    else
    {
        input = input + "8";
    }

    outcome.setText(input);

    Log.v("my.msg", "onBtn8");
}

public void onBtn9(View v)
{
    String input = outcome.getText().toString();

    if(input.equals("0"))
    {
        input = "9";
    }

    else if(input.length() >=15 )
    {
        input = input;
    }

    else
    {
        input = input + "9";
    }

    outcome.setText(input);

    Log.v("my.msg", "onBtn9");
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    outcome = findViewById(R.id.Text_Rlt);
    outcome.setText("0");
}

@Override
public boolean onTouchEvent(MotionEvent event) {

```

```

float x = event.getX();
float y = event.getY();

int act = event.getAction();
{
if(act == MotionEvent.ACTION_DOWN)
{
//Toast.makeText(getApplicationContext(),""+x+"", "+y, Toast.LENGTH_LONG).show();
if( x>=10 && x<=840 && y>=260 && y<=380 )
{
Toast.makeText(getApplicationContext(), "[김지원- 20181151]",
Toast.LENGTH_LONG).show();
}
}
}
return super.onTouchEvent(event);
}
}

```

② 소스 코드 분석

<pre> package com.example.week5_cal; import androidx.appcompat.app.AppCompatActivity; import android.os.Bundle; import android.util.Log; import android.view.MotionEvent; import android.view.View; import android.widget.TextView; import android.widget.Toast; public class MainActivity extends AppCompatActivity { TextView outcome; float mArg; int state; float result; </pre>	<h3>1. MainActivity class</h3> <ul style="list-style-type: none"> - outcome, mArg, state, result의 함수를 지정한다.(mArg와 result는 실수의 값이 나올 수 있어 int형이 아닌 float형으로 선언) - outcome의 값은 TextView에 출력된다. - mArg는 사칙연산에서 연산기호 전에 입력한 수의 값을 저장하는 변수이다. - state는 사칙연산 시 연산기호의 상태를 임의의 숫자로 표현한다.(+:1, -:2, x:3, /:4) - result는 계산된 값을 저장하는 변수이다.
<pre> public void onBtnDel(View v) { String input = outcome.getText().toString(); input = input.substring(0,input.length()-1); outcome.setText(input); if(input.equals("")) { outcome.setText("0"); } Log.v(tag: "my.msg", msg: "onBtnDel"); } </pre>	<h3>2. onBtnDel method</h3> <ul style="list-style-type: none"> - 출력된 값의 가장 낮은 자릿수부터 한 자릿씩 제거하는 Button을 구현한다. - substring으로 마지막 한 자릿수를 제외한 문자열을 input으로 저장한다. - 만약 출력값이 비어있으면 숫자 0을 출력한다. - Logcat을 이용하여 Button의 동작을 확인한다.





<pre> public void onBtnAC(View v) { outcome.setText("0"); Log.v(tag: "my.msg", msg: "onBtnAC"); } </pre>	<p>3. onBtnAC method</p> <ul style="list-style-type: none"> - 출력된 값을 초기화하는 Button을 구현한다. - Button을 클릭하면 출력값이 초기값인 0으로 수정된다. - Logcat을 이용하여 Button의 동작을 확인한다.
<pre> public void onBtnOFF(View v) { finish(); Log.v(tag: "my.msg", msg: "onBtnOFF"); } </pre>	<p>4. onBtnOFF method</p> <ul style="list-style-type: none"> - Button을 클릭하면 앱이 종료되게 구현한다. - finish()를 이용하여 앱을 종료시킨다. - Logcat을 이용하여 Button의 동작을 확인한다.
<pre> public void onBtnSqrt(View v) { String str = outcome.getText().toString(); mArg = Float.valueOf(str); String check = Double.toString(Math.sqrt(mArg)); if(check.length() > 15) { outcome.setText(String.format("%.12f",Math.sqrt(mArg))); } else if(check.length() <= 15) { outcome.setText(""+Math.sqrt(mArg)); } if(Math.sqrt(mArg)-Math.round(Math.sqrt(mArg)) == 0) { outcome.setText(""+Math.round(Math.sqrt(mArg))); } Log.v(tag: "my.msg", msg: "onBtnSqrt"); } </pre>	<p>5. onBtnSqrt method</p> <ul style="list-style-type: none"> - 숫자를 입력한 뒤 Button을 클릭하면 해당 숫자의 제곱근이 출력된다. (Math.sqrt() 이용) - 제곱근 수의 총 길이가 16자리 이상이면 소수점 12번째 자리에서 반올림한다. (String.format() 이용) - 제곱근이 정수로 딱 떨어지는 숫자이면 소수점 아래의 숫자는 제거하도록 소수점 아래 숫자는 버림한다.(Math.round() 이용) - Logcat을 이용하여 Button의 동작을 확인한다.
<pre> public void onBtnSin(View v) { String str = outcome.getText().toString(); mArg = Float.valueOf(str); String check = Double.toString(Math.sin(mArg)); if(check.length() > 15) { outcome.setText(String.format("%.12f",Math.sin(mArg))); } else if(check.length() <= 15) { outcome.setText(""+Math.sin(mArg)); } if(Math.sin(mArg)-Math.round(Math.sqrt(Math.sin(mArg))) == 0) { outcome.setText(""+Math.round(Math.sin(mArg))); } Log.v(tag: "my.msg", msg: "onBtnSin"); } </pre>	<p>6. onBtnSin method</p> <ul style="list-style-type: none"> - 숫자를 입력한 뒤 Button을 클릭하면 해당 숫자의 sin값이 출력된다.(Math.sin() 이용) - sin 값의 총 길이가 16자리 이상이면 소수점 12번째 자리에서 반올림한다. (String.format() 이용) - sin 값이 정수로 딱 떨어지는 숫자이면 소수점 아래의 숫자는 제거하도록 소수점 아래 숫자는 버림한다.(Math.round() 이용) - Logcat을 이용하여 Button의 동작을 확인한다.

<pre> public void onBtnCos(View v) { String str = outcome.getText().toString(); mArg = Float.valueOf(str); String check = Double.toString(Math.cos(mArg)); if(check.length() > 15) { outcome.setText(String.format("%.12f", Math.cos(mArg))); } else if(check.length() <= 15) { outcome.setText(""+Math.cos(mArg)); } if(Math.cos(mArg)-Math.round(Math.sqrt(Math.cos(mArg))) == 0) { outcome.setText(""+Math.round(Math.cos(mArg))); } Log.v(tag: "my.msg", msg: "onBtnCos"); } </pre>	<p>7. onBtnCos method</p> <ul style="list-style-type: none"> - 숫자를 입력한 뒤 Button을 클릭하면 해당 숫자의 cos값이 출력된다.(Math.cos() 이용) - cos 값의 총 길이가 16자리 이상이면 소수점 12번째 자리에서 반올림한다. (String.format() 이용) - cos 값이 정수로 딱 떨어지는 숫자이면 소수점 아래의 숫자는 제거하도록 소수점 아래 숫자는 버림한다.(Math.round() 이용) - Logcat을 이용하여 Button의 동작을 확인한다.
<pre> public void onBtnTan(View v) { String str = outcome.getText().toString(); mArg = Float.valueOf(str); String check = Double.toString(Math.tan(mArg)); if(check.length() > 15) { outcome.setText(String.format("%.12f", Math.tan(mArg))); } else if(check.length() <= 15) { outcome.setText(""+Math.tan(mArg)); } if(Math.tan(mArg)-Math.round(Math.sqrt(Math.tan(mArg))) == 0) { outcome.setText(""+Math.round(Math.tan(mArg))); } Log.v(tag: "my.msg", msg: "onBtnTan"); } </pre>	<p>8. onBtnTan method</p> <ul style="list-style-type: none"> - 숫자를 입력한 뒤 Button을 클릭하면 해당 숫자의 tan값이 출력된다.(Math.tan() 이용) - tan 값의 총 길이가 16자리 이상이면 소수점 12번째 자리에서 반올림한다. (String.format() 이용) - tan 값이 정수로 딱 떨어지는 숫자이면 소수점 아래의 숫자는 제거하도록 소수점 아래 숫자는 버림한다.(Math.round() 이용) - Logcat을 이용하여 Button의 동작을 확인한다.
<pre> public void onBtnSum(View v) { String str = outcome.getText().toString(); //mArg = Integer.parseInt(str); mArg = Float.valueOf(str); outcome.setText("0"); state = 1; Log.v(tag: "my.msg", msg: "onBtnSum"); } </pre>	<p>9. onBtnSum method</p> <ul style="list-style-type: none"> - 숫자를 입력한 뒤 "+" Button을 누른 후 숫자를 입력하고 "=" Button을 누르면 덧셈 시작연산이 구현된다. - 해당 method에서는 시작연산 이전에 입력된 숫자를 형을 변환하여 mArg에 저장한다. - 버튼을 누르면 TextView에는 "0"이 출력된다. - state는 1로 저장된다.("+")기호의 상태를 의미) - Logcat을 이용하여 Button의 동작을 확인한다.
<pre> public void onBtnMin(View v) { String str = outcome.getText().toString(); //mArg = Integer.parseInt(str); mArg = Float.valueOf(str); outcome.setText("0"); state = 2; Log.v(tag: "my.msg", msg: "onBtnMin"); } </pre>	<p>10. onBtnMin method</p> <ul style="list-style-type: none"> - 숫자를 입력한 뒤 "-" Button을 누른 후 숫자를 입력하고 "=" Button을 누르면 뺄셈 시작연산이 구현된다. - 해당 method에서는 시작연산 이전에 입력된 숫자를 형을 변환하여 mArg에 저장한다. - 버튼을 누르면 TextView에는 "0"이 출력된다. - state는 2로 저장된다.("-")기호의 상태를 의미) - Logcat을 이용하여 Button의 동작을 확인한다.

<pre> public void onBtnMul(View v) { String str = outcome.getText().toString(); //mArg = Integer.parseInt(str); mArg = Float.valueOf(str); outcome.setText("0"); state = 3; Log.v(tag: "my.msg", msg: "onBtnMul"); } </pre>	<p>11. onBtnMul method</p> <ul style="list-style-type: none"> - 숫자를 입력한 뒤 “x” Button을 누른 후 숫자를 입력하고 “=” Button을 누르면 곱셈 사칙연산이 구현된다. - 해당 method에서는 사칙연산 이전에 입력된 숫자를 형을 변환하여 mArg에 저장한다. - 버튼을 누르면 TextView에는 “O”이 출력된다. - state는 3로 저장된다.(“x”기호의 상태를 의미) - Logcat을 이용하여 Button의 동작을 확인한다.
<pre> public void onBtnDiv(View v) { String str = outcome.getText().toString(); //mArg = Integer.parseInt(str); mArg = Float.valueOf(str); outcome.setText("0"); state = 4; Log.v(tag: "my.msg", msg: "onBtnDiv"); } </pre>	<p>12. onBtnDiv method</p> <ul style="list-style-type: none"> - 숫자를 입력한 뒤 “/” Button을 누른 후 숫자를 입력하고 “=” Button을 누르면 나눗셈 사칙연산이 구현된다. - 해당 method에서는 사칙연산 이전에 입력된 숫자를 형을 변환하여 mArg에 저장한다. - 버튼을 누르면 TextView에는 “O”이 출력된다. - state는 4로 저장된다.(“/”기호의 상태를 의미) - Logcat을 이용하여 Button의 동작을 확인한다.
<pre> public void onBtnEq1(View v) { String str = outcome.getText().toString(); //int current = Integer.parseInt(str); float current = Float.valueOf(str); // + click if(state == 1) { result = mArg + current; } // - click if(state == 2) { result = mArg - current; } // x click if(state == 3) { result = mArg * current; } // / click if(state == 4) { result = mArg / current; } if(result-Math.round(result) == 0) { outcome.setText(""+Math.round(result)); } else if(result-Math.round(result) != 0) { Toast.makeText(getApplicationContext(), ""+Float.to if(Float.toString(result).length() > 15) { outcome.setText(String.format("%.12",result)); } else if(Float.toString(result).length() <= 15) { outcome.setText(""+result); } outcome.setText(""+result); } Log.v(tag: "my.msg", msg: "onBtnEq1"); } </pre>	<p>13. onBtnEq1 method</p> <ul style="list-style-type: none"> - 하나의 숫자와 하나의 사칙연산 입력 후 두 번째 숫자를 입력하고 “=” Button을 누르면 사칙연산의 값이 TextView에 출력된다. - 두 번째 숫자는 current 변수에 저장된다. - 각 state에 맞는 사칙 연산을 시행한다. state = 1 : 덧셈 연산 state = 2 : 뺄셈 연산 state = 3 : 곱셈 연산 state = 4 : 나눗셈 연산 - 연산의 결과가 정수로 딱 떨어지는 숫자이면 소수점 아래의 숫자는 제거하도록 소수점 아래 숫자는 버림한다.(Math.round() 이용) - 연산의 결과가 정수로 딱 떨어지는 숫자가 아닐 경우 수의 총 길이가 16자리 이상이면 소수점 12번째 자리에서 반올림한다.(String.format() 이용) - Logcat을 이용하여 Button의 동작을 확인한다.

<pre> public void onBtn0(View v) { String input = outcome.getText().toString(); if(input.equals("0")) { return; } else if(input.length() >=15) { input = input; } else { input = input + "0"; } outcome.setText(input); Log.v(tag: "my.msg", msg: "onBtn0"); } </pre>	<p>14. onBtnO method</p> <ul style="list-style-type: none"> - 출력값이 원래 0이면 0을 입력하여도 그대로 0의 값을 출력한다. - 출력값이 0이 아닌 수 일 경우, 출력값의 뒤에 숫자를 이어서 출력한다. - 만약 출력값의 총 길이가 16자리 이상이면 더 이상 숫자가 입력되지 않는다. - Logcat을 이용하여 Button의 동작을 확인한다.
<pre> public void onBtn1(View v) { String input = outcome.getText().toString(); if(input.equals("0")) { input = "1"; } else if(input.length() >=15) { input = input; } else { input = input + "1"; } outcome.setText(input); Log.v(tag: "my.msg", msg: "onBtn1"); } </pre>	<p>15. onBtn1 ~ onBtn9 method</p> <ul style="list-style-type: none"> - 출력값이 원래 0이면 기존의 0을 제거하고 입력값을 출력한다. - 출력값이 0이 아닌 수 일 경우, 출력값의 뒤에 숫자를 이어서 출력한다. - 만약 출력값의 총 길이가 16자리 이상이면 더 이상 숫자가 입력되지 않는다. - Logcat을 이용하여 Button의 동작을 확인한다. - 1 입력 Button부터 9 입력 Button 까지 모두 동일한 방식으로 작동된다.
<pre> @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); outcome = findViewById(R.id.Text_Rlt); outcome.setText("0"); } </pre>	<p>16. onCreate method</p> <ul style="list-style-type: none"> - 앱 실행시 가장 먼저 실행되는 method로 TextView를 활성화 시키고 앱 실행 시, TextView의 출력값을 "0"으로 고정시킨다.
<pre> @Override public boolean onTouchEvent(MotionEvent event) { float x = event.getX(); float y = event.getY(); int act = event.getAction(); { if(act == MotionEvent.ACTION_DOWN) { //Toast.makeText(getApplicationContext(),"xx",Toast.LENGTH_LONG).show(); if(x>10 && x<640 && y>200 && y<300) { Toast.makeText(getApplicationContext(), "id: [권지환 - 20181151]", Toast.LENGTH_LONG).show(); } } } return super.onTouchEvent(event); } </pre>	<p>17. onTouchEvent method</p> <ul style="list-style-type: none"> - 출력값이 표시되는 TextView의 창을 클릭할 경우 Toast로 학번과 이름이 출력된다.

3) 결과 화면

	
<p>- 앱 실행 시 초기 화면</p>	<p>- 연산 계산 후 숫자 출력 화면</p>
	
<p>- TextView 클릭 시 Toast 출력 화면</p>	<p>- OFF Button 클릭 시 앱 종료 화면</p>

3. 결론

1) 나만의 계산기의 장점

① 결과 값이 정수인지 실수인지 구분하여 출력하였다.

: 단순 덧셈이나 뺄셈과 같은 사칙 연산에서는 항상 결과가 정수로 나오지만, 나눗셈이나 제곱근 등의 연산에서는 정수로 나누어 떨어지지 않는 실수의 숫자가 연산의 결과로 나오기도 한다. 따라서 출력값의 변수의 형태를 float로 지정하고, 출력값을 나타낼 때마다 정수인지 실수인지를 판단하여 정수일 경우 소수점 아래 숫자를 제외한 정수 부분만 출력되고, 실수일 경우 소수점 아래 숫자까지 출력되게 하여 실제 계산기와의 유사성을 높였다.

② sqrt, sin, cos, tan 연산 시, “=” Button을 클릭 과정을 생략하여도 바로 결과 값이 출력되도록 제작하였다.

: 일반 계산기에서도 연산자와 피연산자 즉, 두 개의 숫자가 필요한 경우에는 enter 혹은 “=” Button을 이용하여 결과값을 도출해내지만 sqrt, sin, cos, tan과 같은 연산 기호는 하나의 숫자로만 연산한다. 따라서 숫자 입력 후 이 4가지의 Button 중 한 가지를 클릭한다면 바로 결과 값이 TextView에 출력된다.

③ 소수점 아래 12번째 자리까지만 출력된다.

: 연산을 하다 보면 소수점 아래 숫자가 길게 나올 때가 있다. 이럴 경우에 소수점 아래 숫자가 12자리가 넘어가면 TextView의 크기가 변화한다.(TextView의 height를 wrap_contents로 지정) 그리고, 실제 계산기도 일정한 길이가 넘어가면 더 이상 입력되지 않는다. 그와 동일하게 연산의 결과가 소수점 아래 12자리 이상이면 반올림하여 출력하고, 그 후에 더 이상 숫자를 입력하지 못하게 구현하였다.

2) 나만의 계산기의 단점

① 입력하는 값이 정수로 한정되어 있다.

: 초기에 디자인할 때 출력 값이 실수로 나오는 경우는 생각하지 못하고 코드를 작성할 때 이 부분을 보완하였다. 코드를 완성시킨 후에 출력 값은 실수가 가능한 반면에, 입력 값은 정수로 한정되어있다. 이 부분을 수정하기 위해 UI의 디자인부터 다시 제작해야 하기 때문에 수정하지 못했다. 다시 제작한다면 ‘.’ Button을 추가하여 실수도 입력가능하게 된다면 더 좋은 기능의 계산기를 만들 수 있다.

3) 배운 점

이번에 계산기를 제작하면서 가장 어려웠던 점은 변수의 자료형을 다루는 부분이었다. 연산 결과가 실수가 나오는 경우를 생각하지 못하고 모든 변수를 int형으로 지정해줬기 때문에 처음 나눗셈 연산 시 '1/2'의 값이 0.5가 아닌 0이 출력되었다. 이 점을 보완하기 위해 float형으로 수정했지만, 다음에 발생한 문제는 Math,을 사용하는 sqrt, sin, cos, tan 연산이 결과 값이 float가 아닌 double형으로 리턴되기 때문에 에러가 발생하였다. 또한, 이 과정에서 실수형과 문자형으로 전환하는데 오류가 발생하였다. 검색을 통해 잘못된 점을 알고 오류를 수정할 수 있었다.

그 다음에는 Button +, -, x, / 중 어떤 Button을 클릭하였는지 알기 위해 state를 사용하는 것에서 고민을 하였다. 상태 표현을 직관적으로 하고 싶었지만 string으로 표현 시 오류가 발생하고 그 오류를 수정하는 방법을 찾지 못해 int형의 state를 사용하였다.

해결을 한 오류도 있고 해결을 하지 못해 다른 방법을 사용한 경우도 있었다. 다른 방법을 고안해 내서 사용한 것이 잘못된 방법은 아니라고 생각하지만 더욱 직관적이고, 좋은 코드를 만들기 위해 더 학습이 필요하다고 느꼈다.