

[자바 프로젝트]

날짜: 2020.05.10 | 작성자: 정시원 | Version: 1.0

1. 요구사항 분석 (Requirement Analysis)

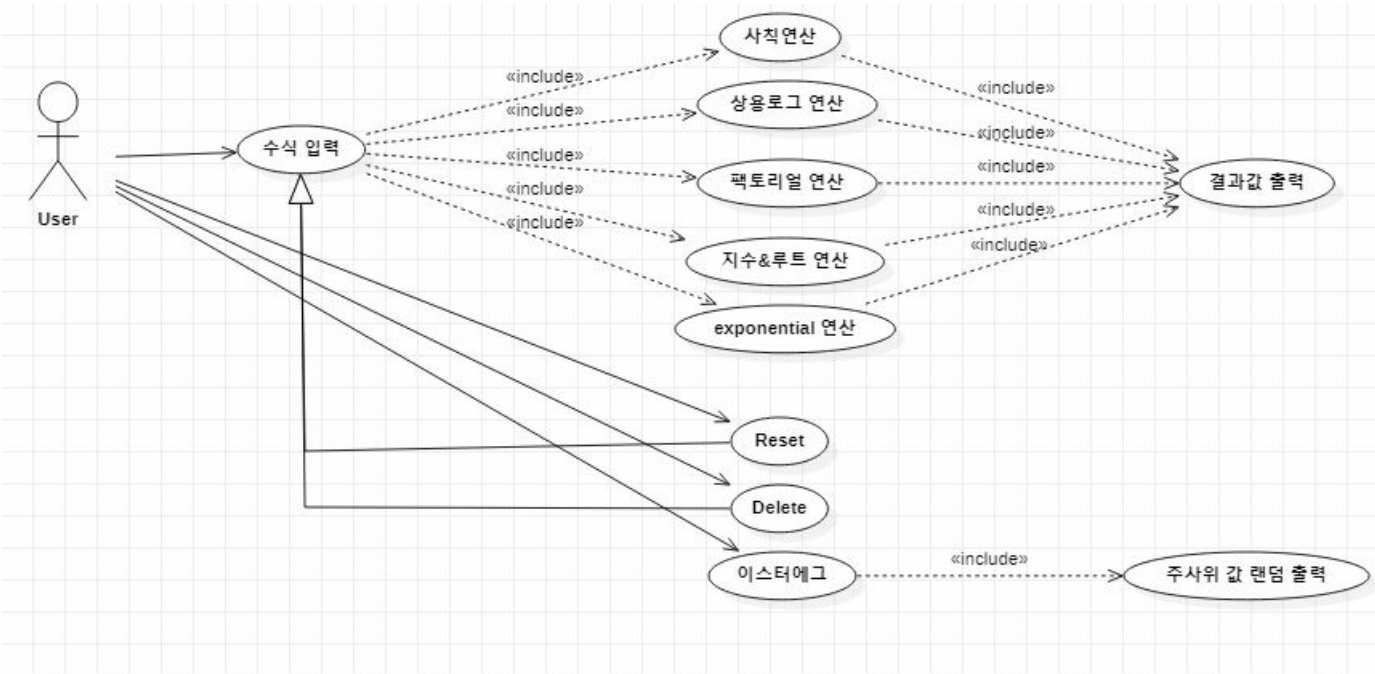
- 공학용 계산기 구현
 - Android Studio 사용
 - 덧셈, 뺄셈, 곱셈, 나눗셈, 나머지, 로그, 지수, 제곱, 루트, 팩토리얼 연산 구현
 - 덧셈, 뺄셈, 곱셈, 나눗셈, 나머지, 로그, 지수, 제곱, 루트, 팩토리얼 연산 구현
 - 실수 연산 구현
 - RESET, DICE(RANDOM INT), DEL 버튼 구현

* "=" 클릭 시 결과 및 연산 식 출력

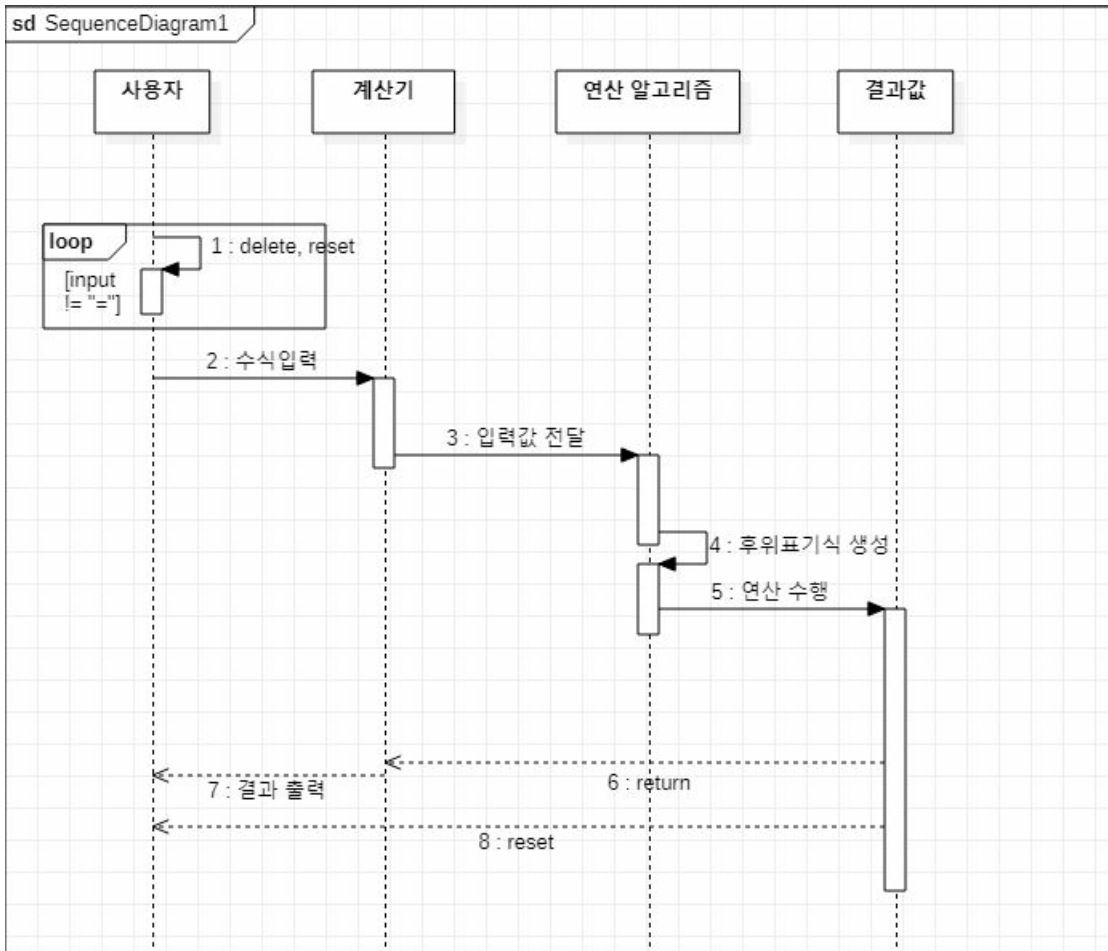
2. 구조 설계

1) UML(Unified Modeling Language) Diagram

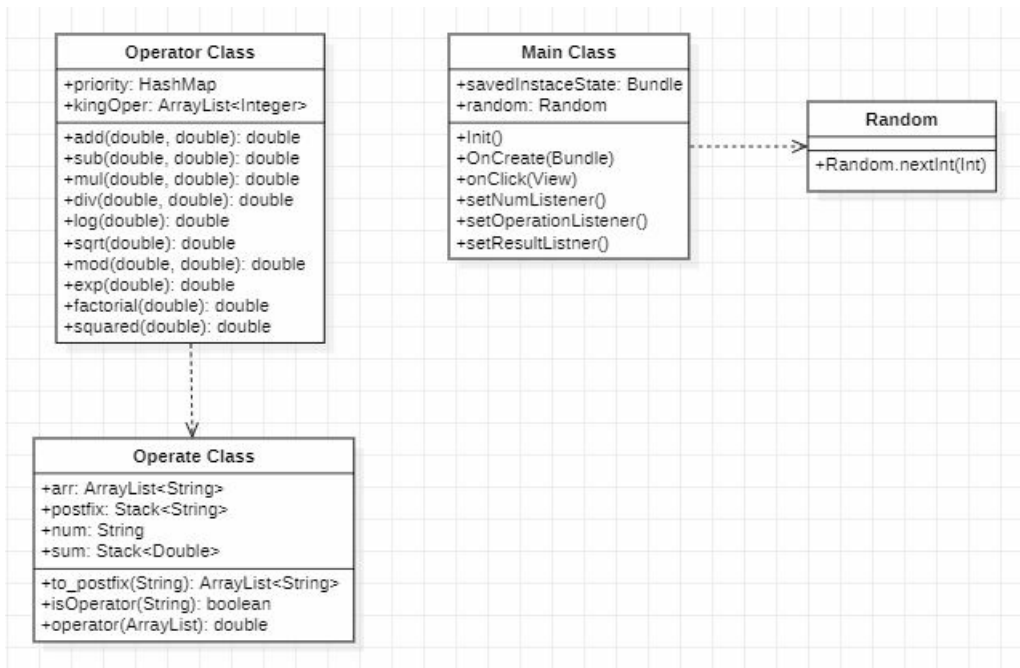
(1) Use Case Diagram



(2) Sequence Diagram



(3) Class Diagram



2) 클래스 정의

- Operator Class
 - 필드 : 연산자 우선순위 HashMap
 - 메소드 : 실수형 연산자(+, -, *, /, %, log, exp, !, ^, √)
- Operate_Process Class
 - 필드 : 변환한 후위표기식 ArrayList, 연산자 Stack, 피연산자 String, 연산결과 Stack
 - 메소드 : 연산자 검사(isOperator), 후위표기식 변환(to_postfix), 후위표기식 연산(operate)
- MainActivity Class

- 필드 : 버튼 위젯 id 선언, 텍스트뷰 위젯 id 선언, 입력 수식 String, 결과값 Double, 출력 수식 String, Operate_Process 클래스 선언
- 메소드 : 생성자 메소드(onCreate), 초기화 메소드(init), 리스너메소드(setNumListener, setOperationListener, setResultListener)

3) 기능

- Operator Class
 - plus 메소드 : 입력된 숫자 2개를 사용하여 덧셈
 - sub 메소드 : 입력된 숫자 2개를 사용하여 뺄셈
 - mul 메소드 : 입력된 숫자 2개를 사용하여 곱셈
 - div 메소드 : 입력된 숫자 2개를 사용하여 나눗셈
 - mod 메소드 : 입력된 숫자 2개를 사용하여 나머지계산
 - squared 메소드 : 입력된 숫자 2개를 사용하여 제곱 계산
 - sqrt 메소드 : 입력된 숫자 1개를 사용하여 루트 계산
 - log 메소드 : 입력된 숫자 1개를 사용하여 상용로그 계산
 - exp 메소드 : 입력된 숫자 1개를 사용하여 지수 계산
 - factorial 메소드 : 입력된 숫자 1개를 사용하여 팩토리얼 계산
- Operate_Process Class
 - isOperator 메소드 : 입력된 문자가 HashMap에 저장되어 있는 연산자인지 아닌지 판별
 - to_postfix 메소드 : 입력된 수식(String)을 연산자 Stack과 피연산자 String을 통해 후위표현식 ArrayList으로 변환
 - operate 메소드 : 입력된 후위표현식(ArrayList)에서 연산자에 따라 연산 진행
- MainActivity Class
 - @Override onCreate 메소드 : 생성한 Button의 리소스 연결을 위해 초기화, 리스너 메소드 호출
 - init 메소드 : Button을 초기화 후 각각의 기능과 연결, Operate_Process 객체 생성, 입력 문자열 input(String)과 입력 수식 operExpression(String)과 연산결과 result(Double) 변수를 초기화한다.
 - setNumListener 메소드 : 숫자 리스너 정의
 - @Override onClick 메소드 : 숫자 버튼이 눌리면 Button id값에 따라 숫자(0~9, ., random)를 operExpression(출력수식)과 input(입력수식)에 문자열로 저장한다. 이때 저장된 연산식이 화면 결과창에 출력되도록 한다. 각 Button 객체의 setOnClickListener 함수를 통해 이벤트 리스너를 등록한다.
 - setOperation 메소드 : 연산자 and del, reset 리스너 정의
 - @Override onClick 메소드 : 연산자 버튼이 눌리면 Button id값에 따라 출력연산자(+, -, *, /, %, log, exp, ^, √, !)를 operExpression(출력수식)에 저장하고, 입력연산자(+, -, *, /, %, !, e, ^, √, !)를 input(입력수식)에 문자열로 저장한다. reset버튼을 누르면 해당 수식이 모두 없어지고 결과창에 0이 출력되도록 한다. del버튼을 누르면 가장 최근에 클릭한 Button의 값을 삭제한다. 이때 저장된 연산식이 화면 결과창에 출력되도록 한다. 각 Button 객체의 setOnClickListener 함수를 통해 이벤트 리스너를 등록한다.
 - setResultListener 메소드 : '=' 연산자 정의
 - @Override onClick 메소드 : '=' 버튼이 눌리면 Button id값에 따라 '='을 출력수식 operExpression에 저장하고 수식 텍스트뷰(input_show)에 출력한다. Operate_Process의 함수들을 호출하여 입력수식 input을 전달하여 결과값 result를 결과 텍스트뷰(result_show)에 출력한다. 그 후 input과 operExpression을 초기화한다. '=' Button 객체의 setOnClickListener 함수를 통해 이벤트 리스너를 등록한다.

4)레이아웃

* activity_main.xml : 사용자가 보는 화면으로 출력되는 UI를 나타낸다. TextView 위젯, Button 위젯을 포함한다.

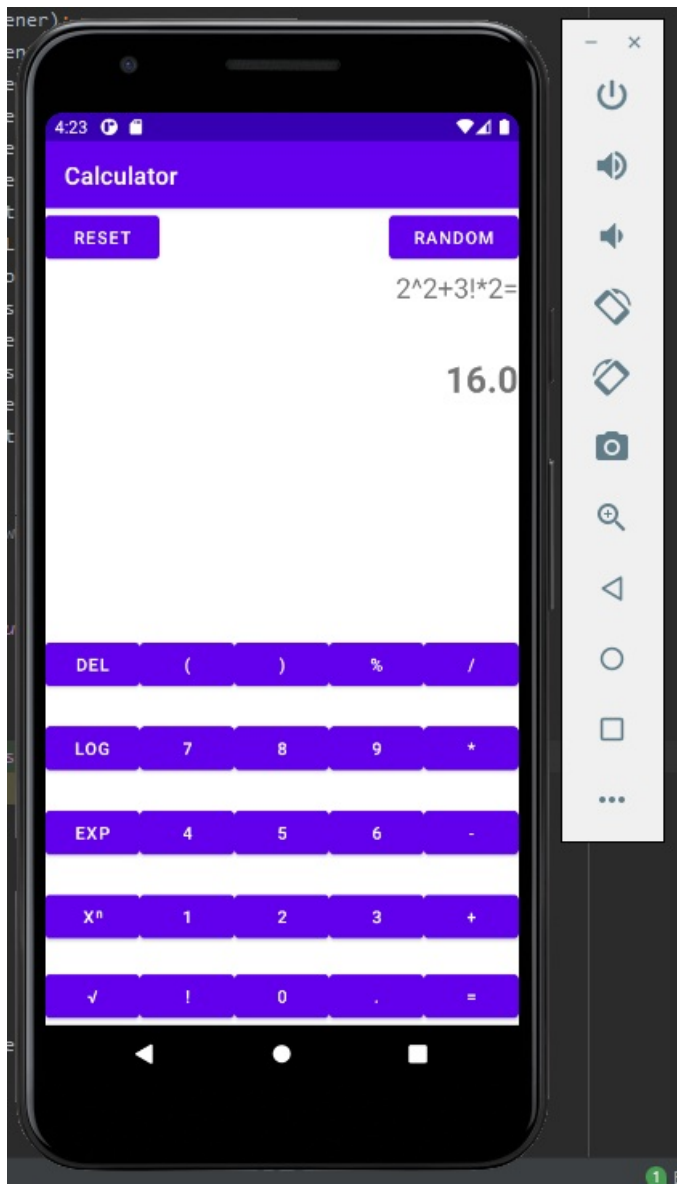
3. 입력/출력 데이터 정의 (Data Structure Design)

- 입력 : Button 위젯(연산자, 피연산자, reset, del)

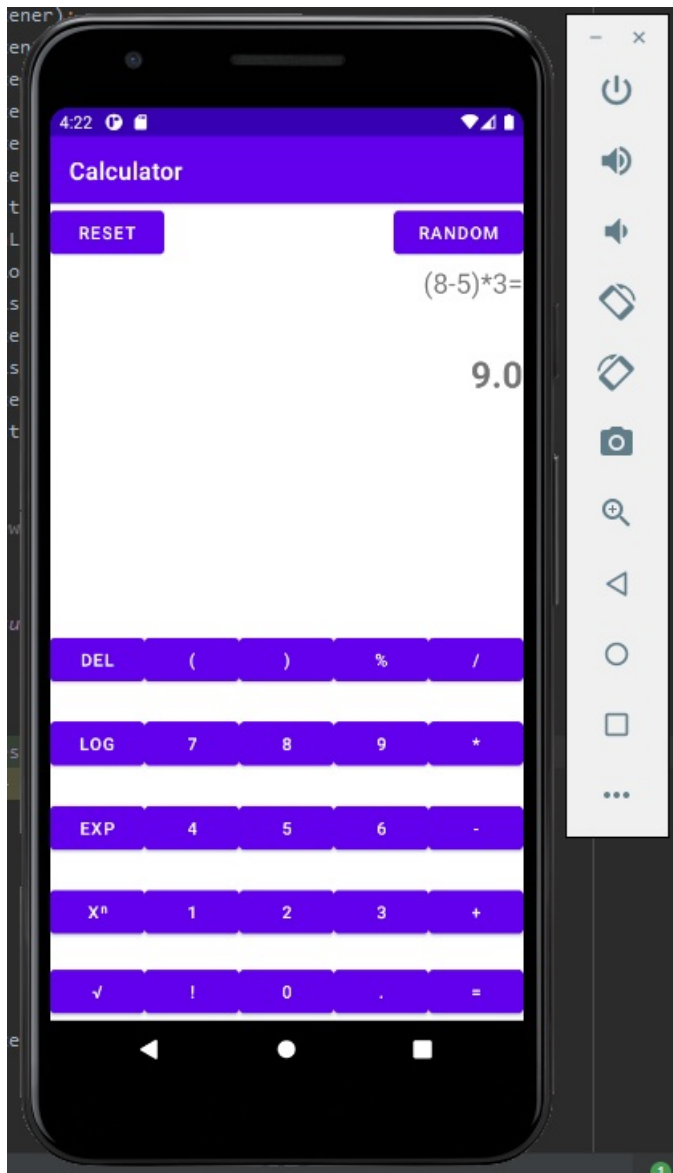
* 출력 : TextView 위젯(수식 및 연산결과 String)

4. 테스트 프로토타입 (Test Prototype)

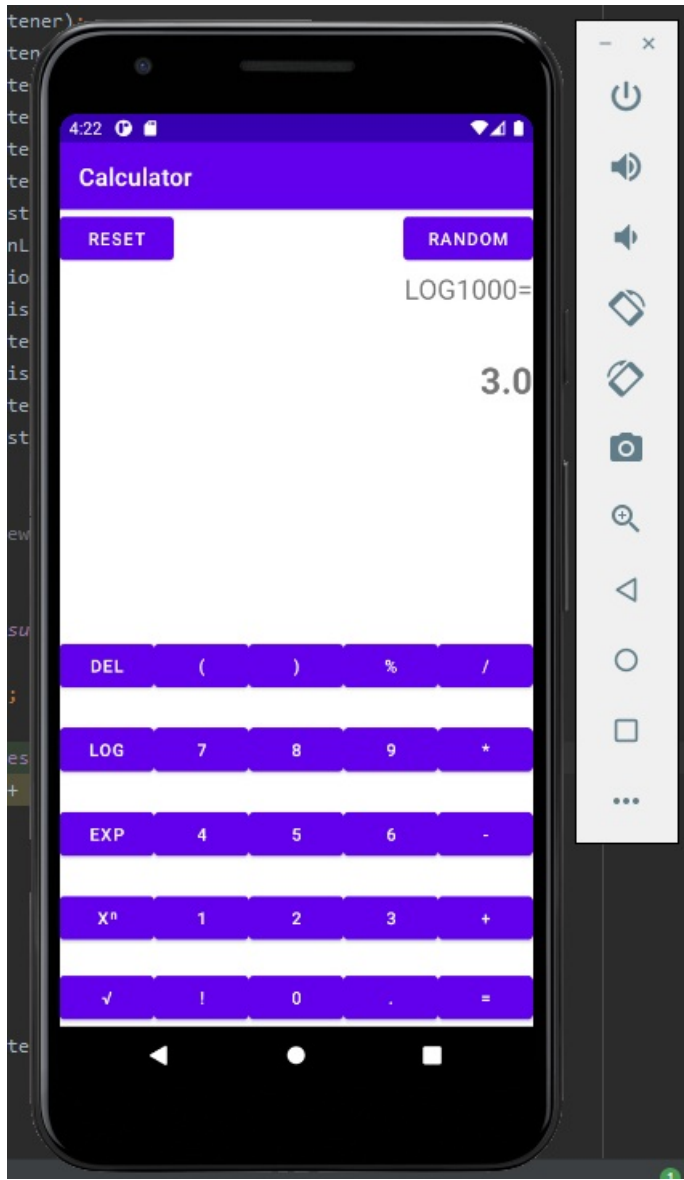
- 사칙 연산 and 팩토리얼 연산



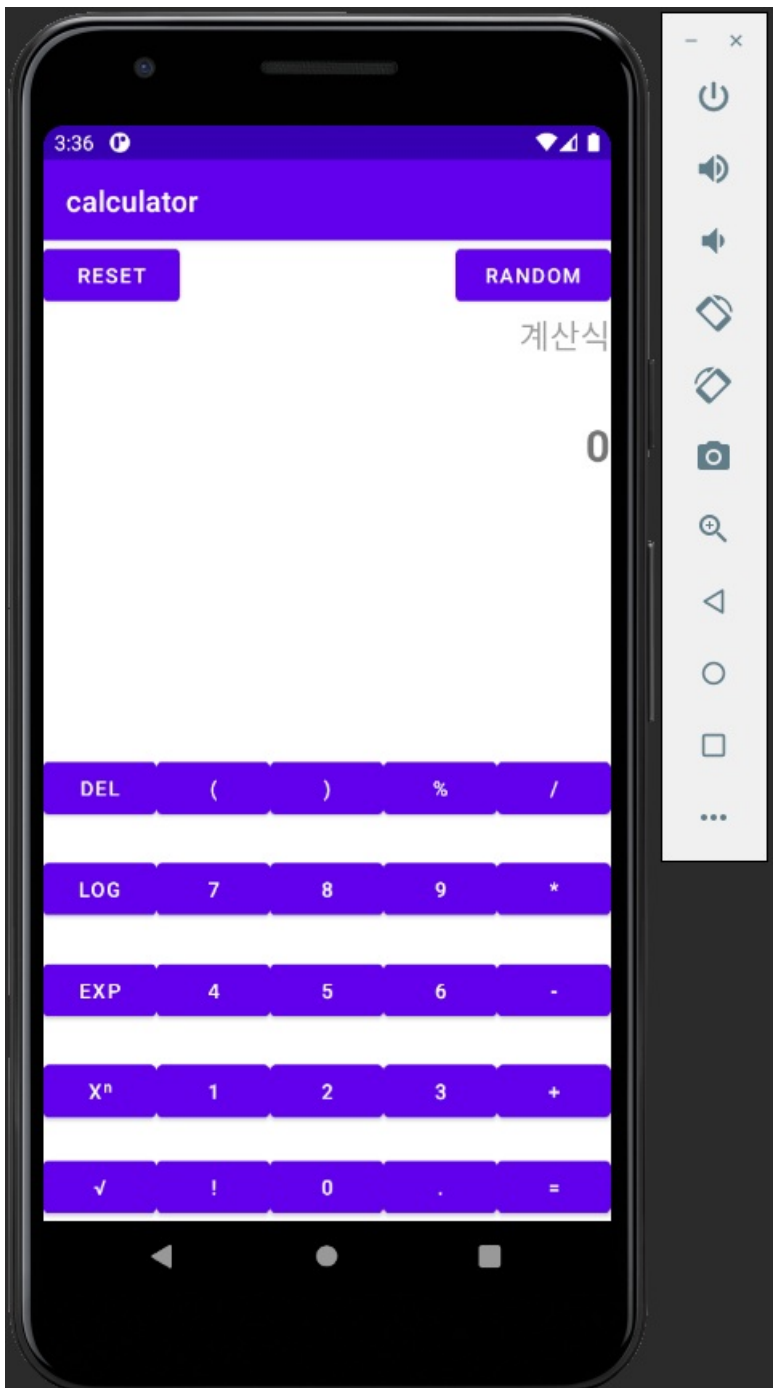
- 괄호 연산



- 상용로그 연산



*버튼 Reset



5. 보완할 점

- 예외처리 필요 : 올바른 수식이 아닐 경우 앱이 강제 종료된다.
- 주요 메소드 알고리즘 순서도 구현
- 스크롤뷰를 통해 연산 HISTORY 기능 추가