R/ E/ P/ O/ R/ T



제출일: 19. 10. 20

과제명: 1차 프로젝트 계산기

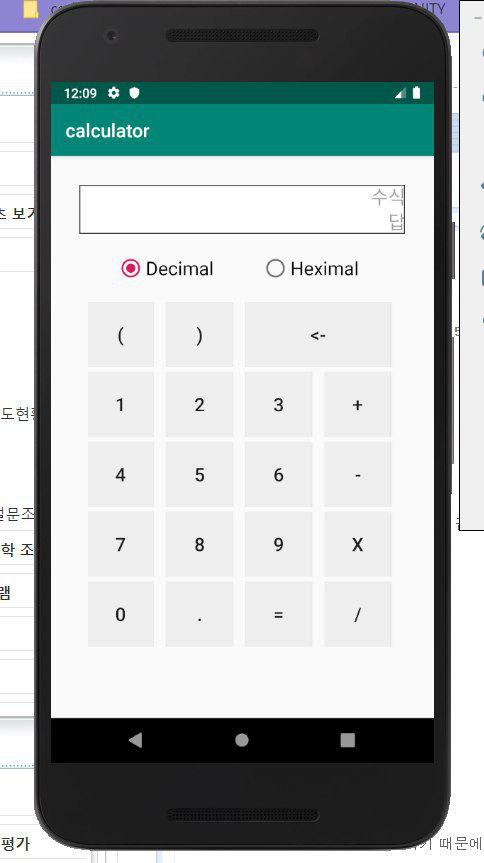
과목 : 심화 전공 실습 1

학과 : 소프트웨어 학부

학번 : 2015726076

이름 : 김현구

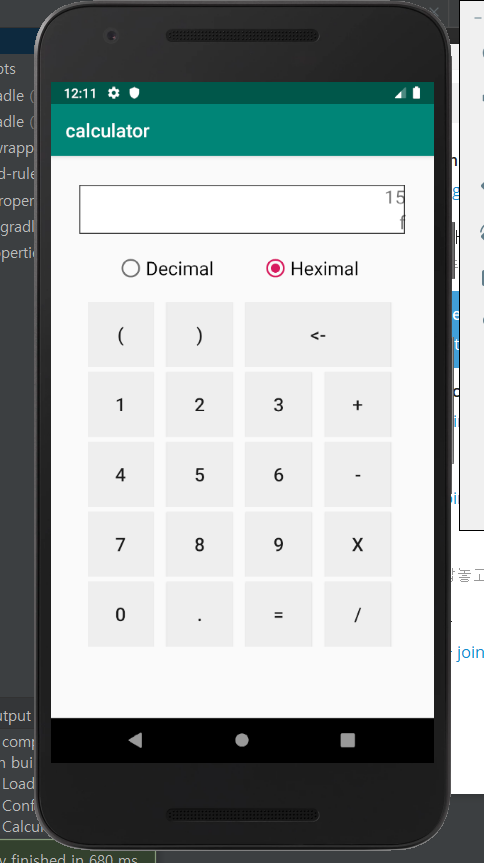
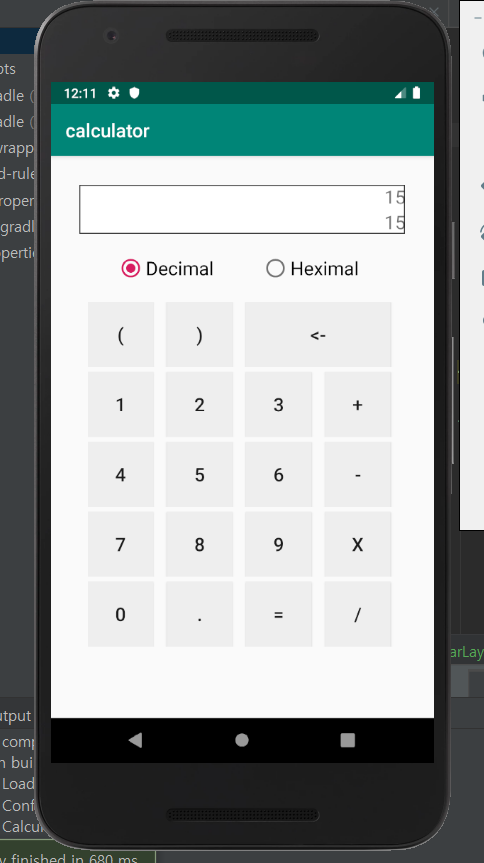
전체 화면



맨 처음 켰을 때 화면으로 Decimal이 체크되어 있다.

세부 구현

1. 진수 변환



15를 16진수로 변환하여 f가 출력되는 모습이다.

<Xml Code> <java Code>



<설명>

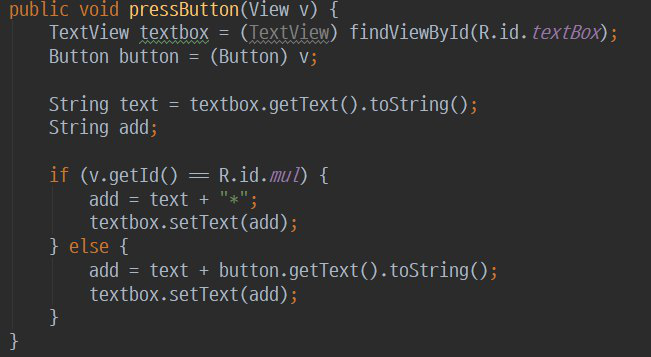
RadioGroup을 통해 가로로 정렬하였고, Decimal 버튼은 checked = “true”로 기본값을 주었다.  
라디오 버튼을 클릭하면 onClick이벤트를 통해 decHex()함수를 호출한다.  
dexHex()에서는 정답을 입력하는 textAnswer박스를 가져와 indexOf함수를 이용해서 소수점을 제거한다.  
R.id를 인식하여 각 버튼의 id를 기준으로 변환을 수행한다.

1. 버튼 클릭



버튼을 누르면 해당 버튼의 문자(X제외)가 위의 textBox에 삽입된다.

<Xml Code> <java Code>

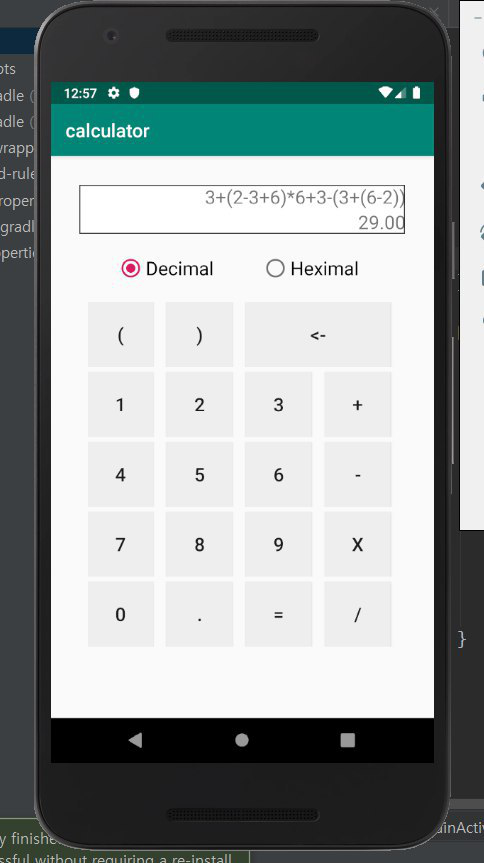


<설명>

TableLayout을 이용해서 4\*5의 계산기를 xml 코드로 만들었고  
각 버튼을 클릭하면 pressButton()함수를 호출하여 id를 통해 그 값이 textbox에 대입하게 된다.

단, 곱하기는 \*문자로 대체하여 삽입하였다.

3. 계산하기

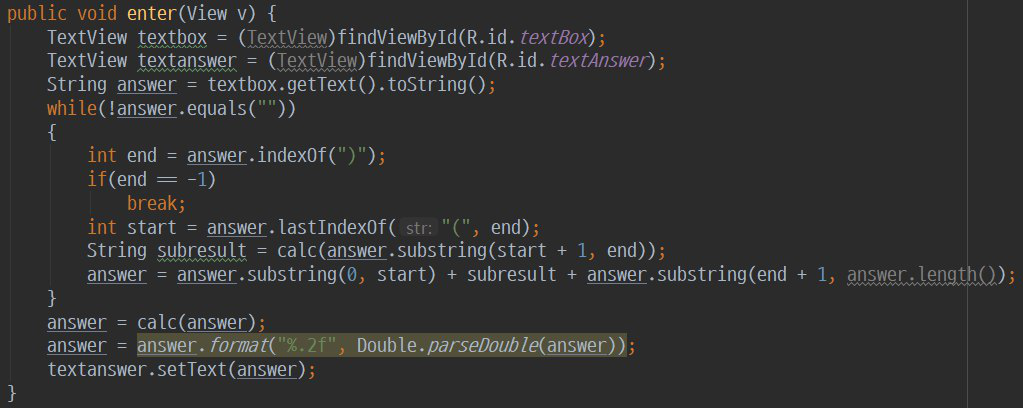


계산은 아래 순서대로 진행된다.

괄호 -> 곱하기, 나누기 및 음수 처리 -> 더하기, 빼기 및 음수 처리

괄호는 enter함수, 그 외 사칙 연산은 calc 함수에서 진행된다.

1) 괄호



<설명>

먼저 전체 문자열을 가져와서 닫는 괄호”)”를 찾고 그 지점부터 역으로 여는 괄호”(“를 찾아서 그 사이의 값을 calc함수에 넣어 계산한다.

Calc 함수의 결과를 괄호를 제외한 이전, 이후 문자열과 결합한다.

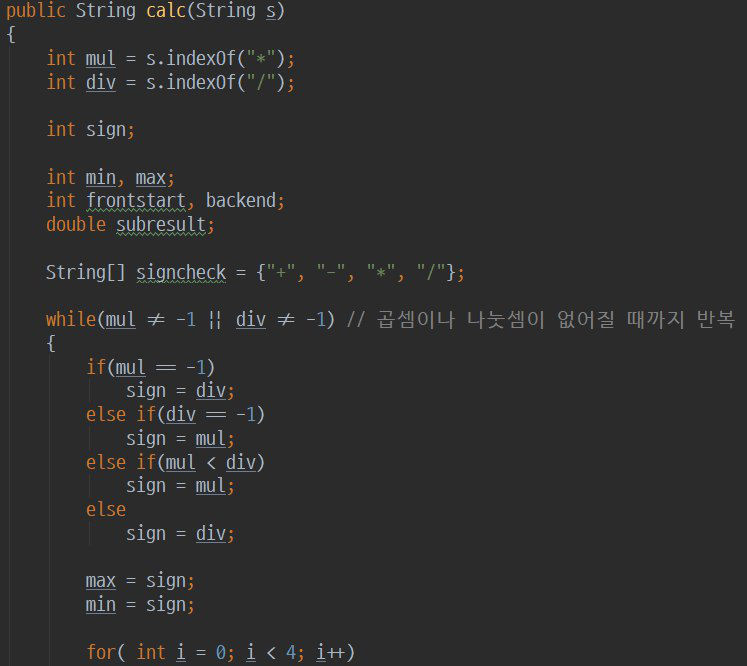
이를 괄호가 사라질 때(end == -1)까지 반복한다.

만약, 괄호가 없다면 while문 밖의 calc함수를 통해 계산된다.

결과는 format %.2f를 통해 소수점 둘 째 자리 까지 출력된다.

2) 곱하기, 나누기

2-1)



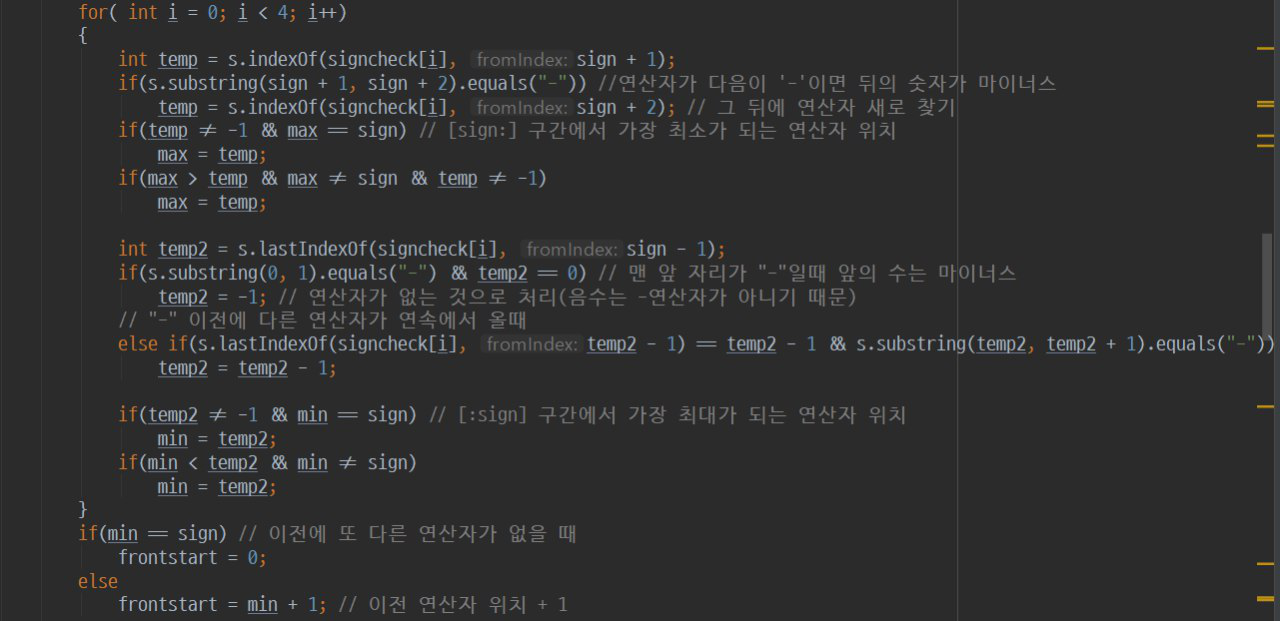
<설명>

변수를 선언하고 while 문을 통해 \*, / 연산이 끝날 때 까지 반복한다.

처음 4개의 if-else 구문으로 부호가 \* 인지 / 인지 판별한다.

그리고 그 부호를 기점으로 좌, 우로 인덱스를 탐색할 max(우), min(좌) 함수를 이용한다.

2-2)



<설명>

Signcheck = {“+”, “-“, “\*”, “/”}배열을 4번 반복하여 탐색하는 반복을 돈다.

반복을 진행하면서 뺄셈 연산이 아닌 마이너스로 사용되는 -를 좌, 우로 판별한다.  
temp – max(우), temp2 – min(좌)

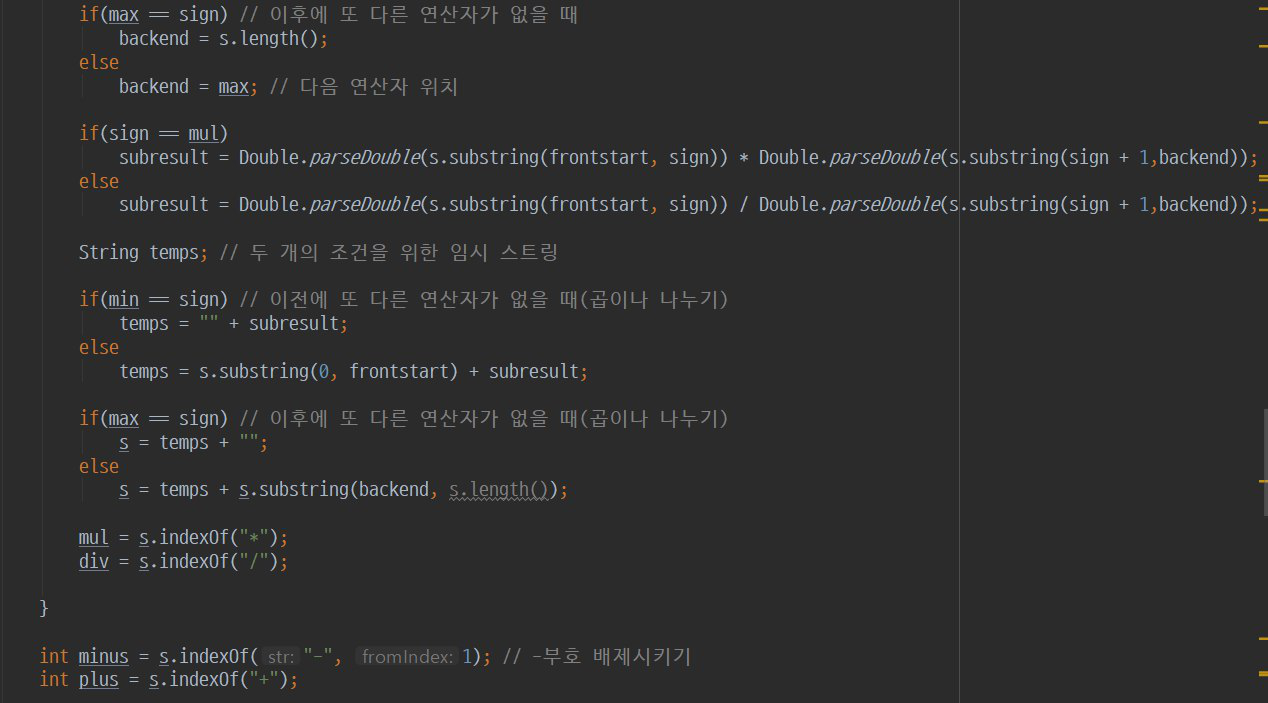
Ex) -16(좌) \* 8(우)

또 후위에서는 연산자가 연속해서 올 경우는 ex) 16\*-8  
-가 반드시 포함되는 경우이므로 예외로 처리하여 그 연산자를 건너뛰고 한번도 indexOf, lastIndexOf함수를 사용하여 그 다음 연산자를 찾는다.

동시에 인덱스를 찾지 못했을 때 -1이 반환되는 것을 유의하여 조건을 진행한다.

결과적으로 앞에 다른 연산자가 있을 경우 min 은 sign보다 작게,  
뒤에 다른 연산자가 있을 경우 max는 sign보다 크게된다. 즉 sign과 다르게 된다.

2-3)



<설명>

이렇게 가운데 곱하기, 나누기 연산자를 기준으로 좌, 우 숫자를 음수를 신경써서 substring으로 뽑아서 계산한다.

Substring의 범위를 벗어나지 않게 string을 합칠 때 조건을 두개로 나누어서 진행한다.

합쳐진 스트링 s에서 다시 곱하기, 나누기 기호를 찾아서 반복한다.

3) 더하기, 빼기





<설명>

곱하기, 나누기 기호가 모두 사라진 이후에 더하기, 빼기 연산을 진행한다.

-는 맨 앞에 붙어서 음수를 가르킬 수 있으므로 1번 인덱스부터 indexOf함수를 사용한다.

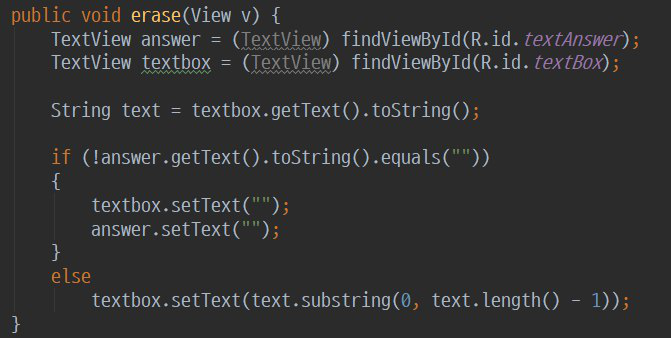
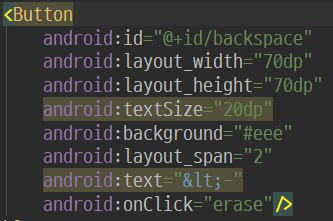
그리고 +, -연산자만 있으므로 순서에 신경쓰지 않고 연산을 하는데, 뒤에 연산자가 있거나 음수가 오는 경우를 신경써서 반복한다.

연산 결과를 string으로 결합하고 반복한다.

모든 반복이 끝나면 계산된 결과를 리턴한다.

4) 지우기

<xml Code> <java Code>



Xml 코드를 보면 &lt; 기호를 이용해 < 를 표시하였고, layout\_span을 2 주어 2의 크기를 가졌다.

정답 텍스트 박스(answer), 식 텍스트 박스(textbox)를 호출하여

정답칸이 비어있으면 둘다 초기화를 시키고, 그렇지 않다면 수식칸의 맨 마지막 수식을 하나 지우게 만들었다.

고찰

스택을 통해 계산을 하는 방식을 잊어서, 인덱스를 참조하는 방식으로 구현을 했는데 계산기가 생각보다 훨씬 복잡하게 동작한다는 것을 새롭게 깨달았다.

또, 자바는 형 변환이 자유롭지 못해 간단한 수식도 복잡하게 길게 늘어뜨려야 하는 점이 불편하게 느껴졌다.