SOEN 6011 : SOFTWARE ENGINEERING PROCESSES
SUMMER 2022

F2: Tangent Function, $tan(x)$

Author

Zeyu Huang

https://github.com/KDKBHZY/Soen6011-ZeyuProject

Contents

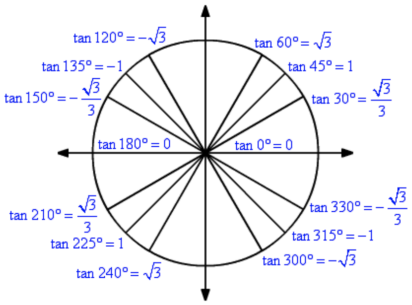
1)Problem1

a)Description of Function

[1] $tan(x)$ is a periodic function which is very important in trigonometry. The simplest way to understand the tangent function is to use the unit circle. For a given angle measure θ draw a unit circle on the coordinate plane and draw the angle centered at the origin, with one side as the positive x -axis. The x -coordinate of the point where the other side of the angle intersects the circle is $cos()$ and the y -coordinate is $sin()$. So, the tangent function is define as below:

$$tan(x) = \frac{sin(x)}{cos(x)}$$

The below graph shows values corresponding to different angles.



[1][2]The tangent function is undefined when $x = \pi\ /\ 2 + n\pi$ (where, n is integer) for which, $cos(x) = 0$. However, Tangent function does not have an amplitude. In addition, The graph intercept x-axis at $n\pi$ (where n is integer) and in y-axis at $(0, 0)$ point. The period of tangent function is π.

Range

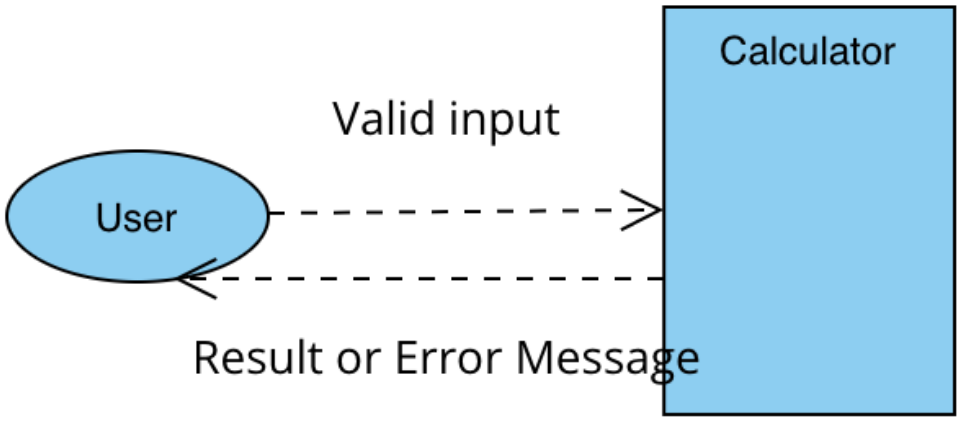
[1][2] The range of $tan(x)$ is all real number $\mathbb{R}$, $(-\infty, +\infty)$.

Domain and Co-domain

[1][2] The domain of tangent function is $x \in \mathbb{R}$, $x \neq \pi\ /\ 2 + n\pi$ where, n is an integer. The co-domain of $tan(x)$ is $(-\infty, +\infty)$.

b)Context of Use Model

Users can use the calculator to calculate the result of $sin()$, $cos()$ and $\frac{sin()}{cos()}$ which is $tan()$ of a degree. This degree shall be an integer or decimal, so the digits *0-9* and the decimal point must be available by the user. The user can select the appropriate function they want to use, and they shall be able to press a button to have the answer computed. The calculator should return the result or an error message that indicates why it was unable to do so.



2)Problem 2

Assumption:

For the given degree x, return the result of $tan(x)$. If the input value is invalid or cannot be calculated, return an error message.

Requirements:

Functional Requirements:

Requirement Id	FR1
Overview	$x = 0° + n\pi$
Description	For the given input $x = 0°$, the function may return 0 as output.
Priority	High
Type	Functional
Difficulty	Easy

Requirement Id	FR2
Overview	x is Positive Degree
Description	For the given input x = any Positive Degree , the function may return corresponding $tan(x)$ value as output.
Priority	High
Type	Functional
Difficulty	Medium

Requirement Id	FR3
Overview	x is Negative Degree
Description	For the given input x = any Negative Degree , the function may return corresponding $tan(x)$ value as output.
Priority	High
Type	Functional
Difficulty	Medium

Requirement Id	FR4
Overview	$x = 90° + n\pi$
Description	For the given input x , the function may return "Invalid" as output.
Priority	High
Type	Functional
Difficulty	Hard

Non-Functional Requirements:

Requirement Id	NFR1
Overview	Maintainability
Description	The ability to add features or fix bugs after the project is finihed. the function may return "Invalid" as output.
Priority	High
Type	NonFunctional
Difficulty	Medium

Requirement Id	NFR2
Overview	Usability
Description	The ability that users can easily use its functions .
Priority	High
Type	NonFunctional
Difficulty	Medium

Requirement Id	NFR3
Overview	Portability
Description	The ability that the project can easily suit in users' system environment .
Priority	High
Type	NonFunctional
Difficulty	High

3)Problem 3

a)Algorithm Selection

For this part, I will introduce two algorithms for implementing $tan(x)$ function. **Polynomial approximation and Maclaurin series.**

Algorithm1 : Polynomial approximation

[3]Polynomial approximation is an approximation of a curve with a polynomial. When we solve mathematical questions, we don't actually know how to calculate certain functions, such as the $sin()$ function. Therefore, to solve this kind of problems, mathematicians develop very good approximations to these functions - related functions which are very close to the function of interest, but much easier to calculate.

Advantages	Disadvantages
Easy to calculate	The approximation is only precise for small x, so some steps are needed when we calculate $tan(x)$

Algorithm2 : Maclaurin series

[4]A Maclaurin series is a Taylor series expansion of a function about 0,

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \frac{f^{(3)}(0)}{3!}x^3 + ... + \frac{f^{(n)}(0)}{n!}x^n$$

The $tan(x)$ function's approximation is derived by the Maclaurin Series's explicit forms of $sin(x)$ and $cos(x)$.

$$sin(x) = x - x^3/3! + x^5/5! - x^7/7! + ... \quad (1)$$

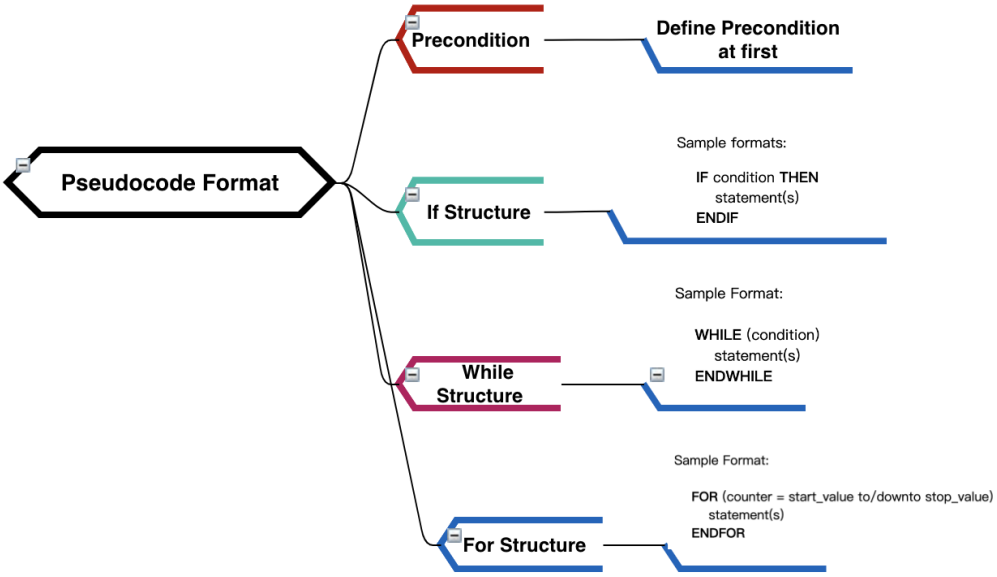
$$cos(x) = 1 - x^2/2! + x^4/4! - x^6/6! + ... \quad (2)$$

Then, we can use $\tan(x) = \frac{sin(x)}{cos(x)}$ to calculate.

Advantages	Disadvantages
The formula $\tan(x) = \frac{sin(x)}{cos(x)}$ is easy to understand.	Successive terms get very complex and hard to derive.

b)Mind Map For Pseudocode

In this part, I will use a mind map to decide a pseudocode format.



c) Pseudocode for each Algorithm

In this part, I will write pseudocode for each algorithm.

Algorithm 1 [5]Polynomial approximation

Require: $x \notin [0°, 180°]$
  function PERIODICITY(x)
    if $x > 180°$ **then**
      while $x > 180°$ **do**
        $x = x - 180°$ ▷ reduce x to the range $[0°, 180°]$
      end while
    else
      while $x < 0°$ **do**
        $x = x + 180°$ ▷ add x to the range $[0°, 180°]$
      end while
    end if
  return x ▷ get valid x
  end function

Require: $x \notin [0°, 90°]$
  function SYMMETRY(x)
    $tan(x) = -tan(180° - x)$ ▷ use the symmetry of $tan()$
  return $tan(x)$ ▷ Return result
  end function

Require: $x \notin [0°, 45°]$
  function COFUNCTION(x)
    $tan(x) = -\frac{1}{tan(90°-x)}$ ▷ use the reciprocal of $tan()$
  return $tan(x)$ ▷ Return result
  end function

Require: $x \notin [0°, 22.5°]$
  function TRIGONOMETRIC_IDENTITY(x)
    $tan(x) = -\frac{2tan(\frac{x}{2})}{1-tan^2(\frac{x}{2})}$ ▷ use the trig identity of $tan()$
  return $tan(x)$ ▷ Return result
  end function

Require: $x \in [0°, 22.5°]$
  function POLYNOMIAL(x)
    x $= x * \frac{\pi}{180°}$ ▷ convert x to radians
    $tan(x) = x + \frac{x^3}{3} + \frac{2x^5}{15} + \frac{17x^7}{315}$ ▷ use the trig identity of $tan()$
  return $tan(x)$ ▷ Return result
  end function

Algorithm 2 Maclaurin Series

Require: x in degrees
  function GETRAD(x)
    $val = x * \frac{\pi}{180°}$ ▷ Calculate x in radians
  return val ▷ return x in radians
  end function

Require: $n \neq 0$ and $length = 0$
  function CHECKDIGITS(n)
    while $n \leqq 1$ **do**
      $length+ = 1$ ▷ Calculate length
      $n* = 10$
    end while
  return $length$ ▷ return length
  end function

Require: $getrad(x) \neq NULL$ AND $n \neq 0$ AND x in radians AND $k = 1$ AND $m = 0$
  function CALCULATESIN(getrad(x))
    $sinres = \frac{x}{k!}$
    while $checkdigits(sinres) \neq n$ **do**
      $k = k + 2$
      if $m\%2 == 0$ **then**
        $sinres- = \frac{x^k}{k!}$
      else
        $sinres+ = \frac{x^k}{k!}$
      end if
      $m+ = 1$
    end while
  return $sinres$ ▷ get value of sin(x)
  end function

Require: $getrad(x) \neq NULL$ AND $n \neq 0$ AND x in radians AND $k = 2$ AND $m = 0$
  function CALCULATECOS(getrad(x))
    $cosres = 1$
    while $checkdigits(cosres) \neq n$ **do**
      $k = k + 2$
      if $m\%2 == 0$ **then**
        $sinres- = \frac{x^k}{k!}$
      else
        $sinres+ = \frac{x^k}{k!}$
      end if
      $m+ = 1$
    end while
  return $cosres$ ▷ get value of cos(x)
  end function

Require: $getrad(x) \neq NULL$ AND $calculatecos(x) \neq NULL$ AND $calculatesin(x) \neq NULL$
  function CALCULATETAN(calculatecos(x),calculatesin(x))
    SinVal = calculatesin(x)
    CosVal = calculatecos(x)
  return $\frac{SinVal}{CosVal}$ ▷ calculation for tan(x)
  end function
  $result \leftarrow tan(x)$

4)Problem 4

a)Debugger

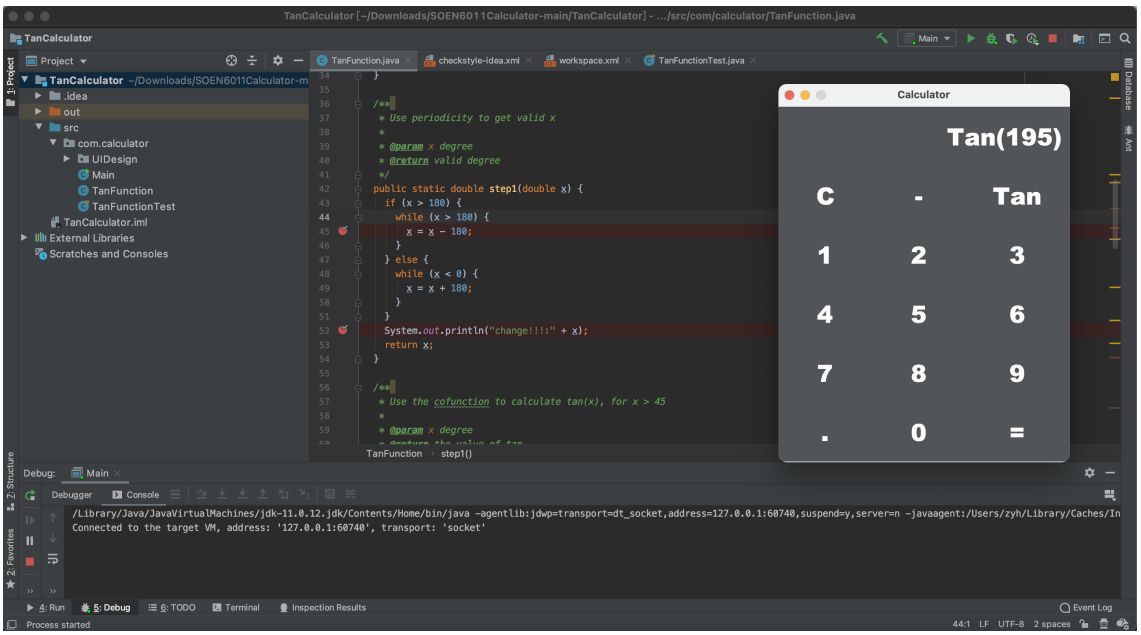
Description:

The debugger I used is the Intellij IDEA built-in debugger. When I click the debug button, it will enter the debug mode. And in the debug mode, it will stop at the break point I set, and can go step by step. In addition to this, I can see the value of variables in each step.

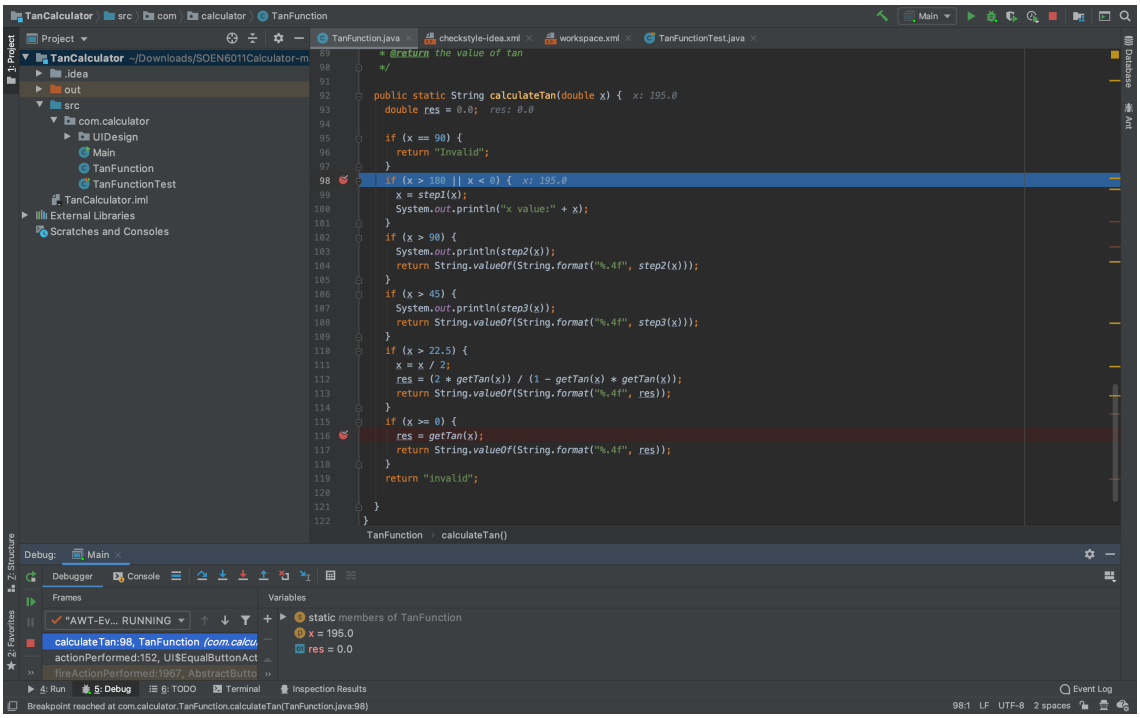
Advantages	Disadvantages
No need to install,easy to use	It may not support multi-threading program
Can see the value of variables in each step	
It provide break point which can stop anywhere	

Example:

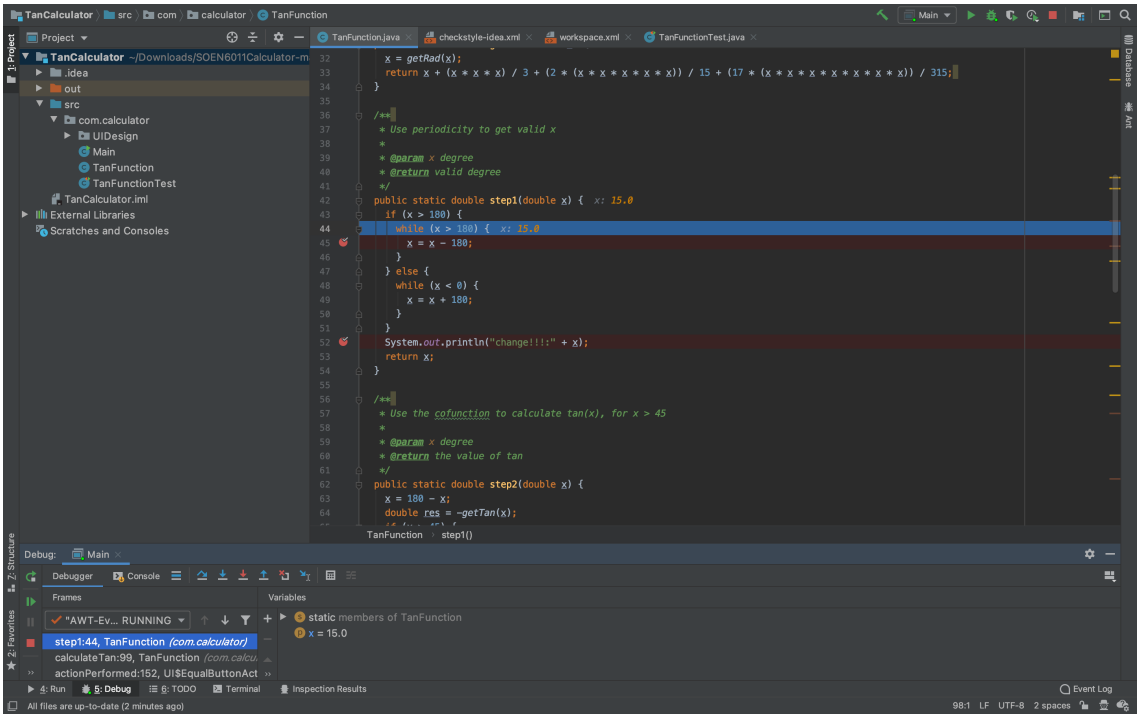
1. Enter a valid input(195°), then press Tan button and "=" button



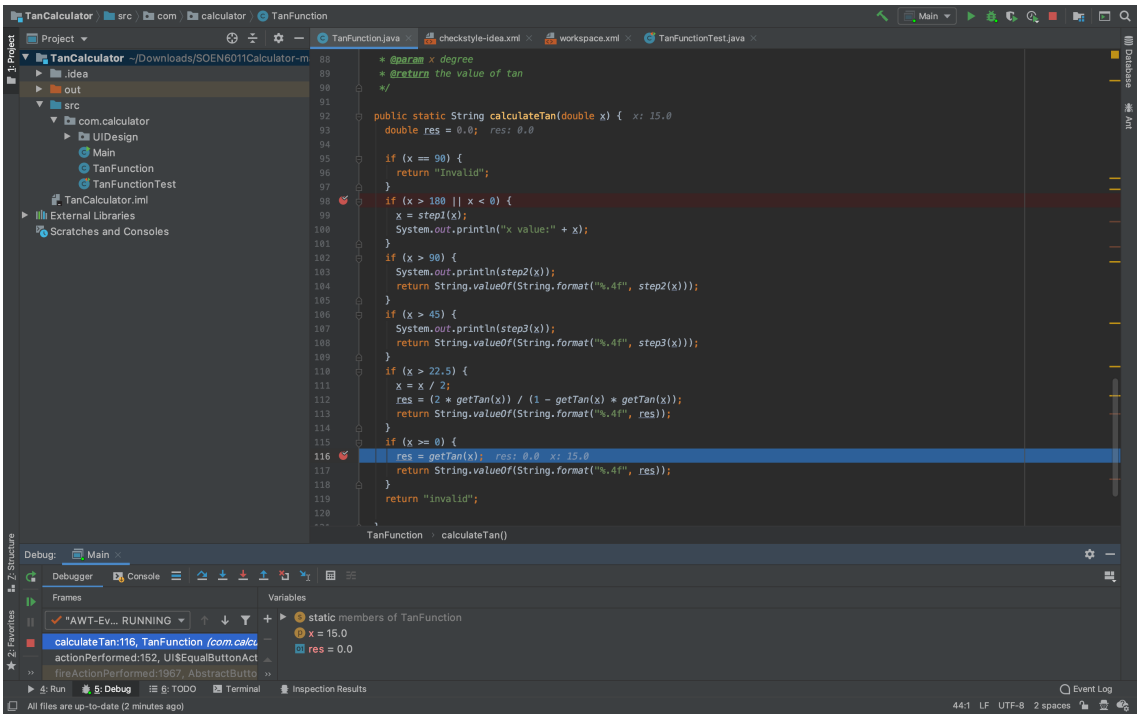
2. Go to calculateTan function and classify the degree.



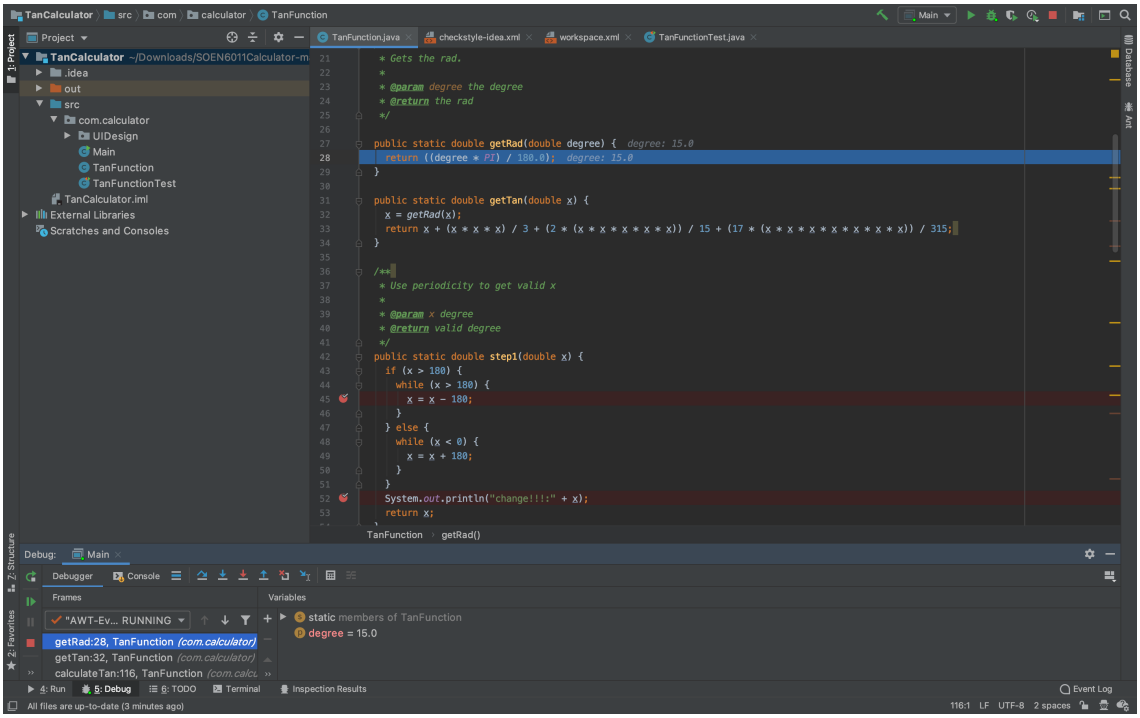
3. Go to step1 function and reduce it to the valid range.(We can see it decrease to 15°)



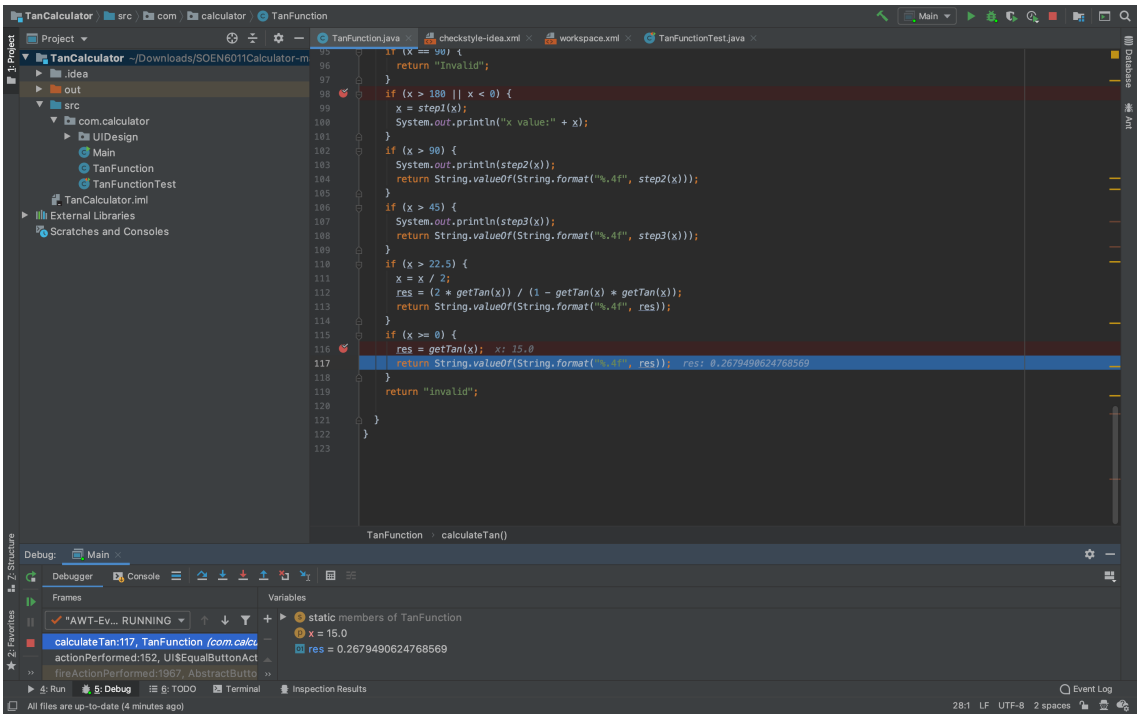
4. Return to calculateTan function and continue judge the degree.

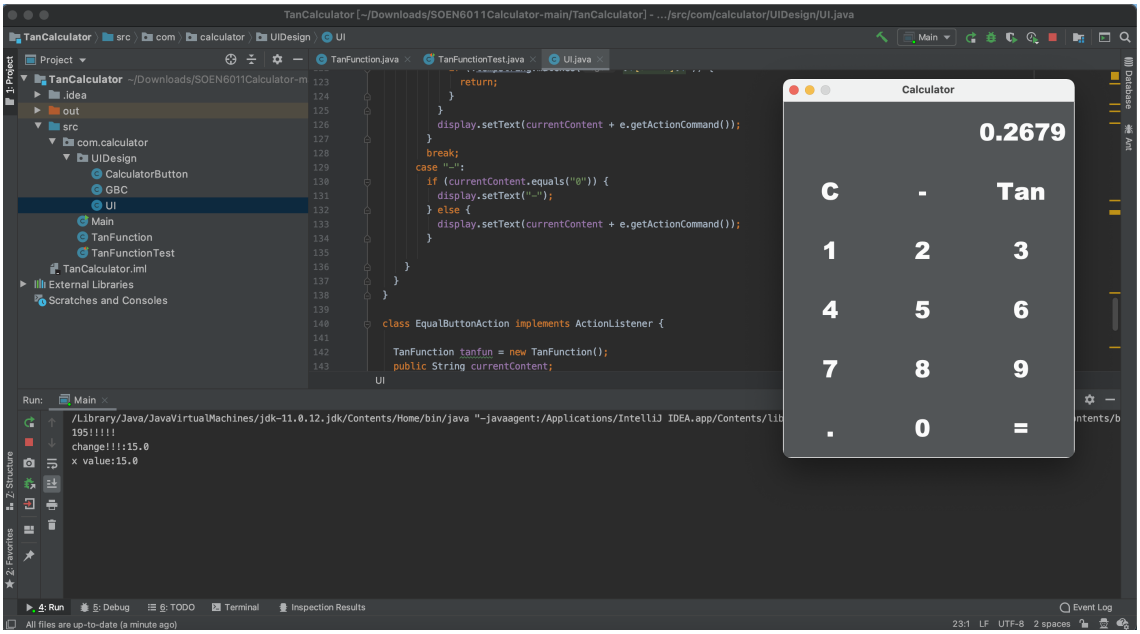


5. Go to getTan function to calculate the result.



6.Get the result and return to UI function and display it on the interface.



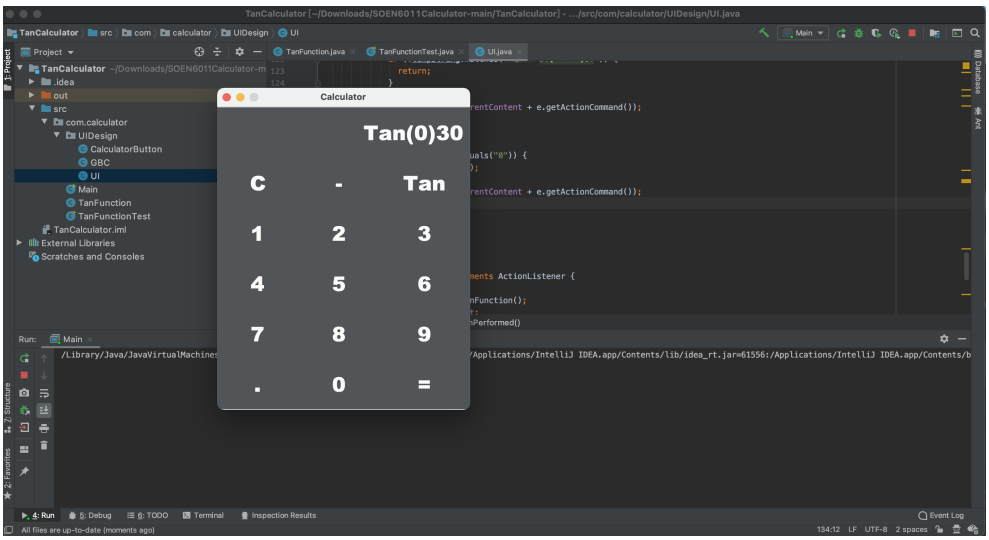


b)Error handling and Error Messaging

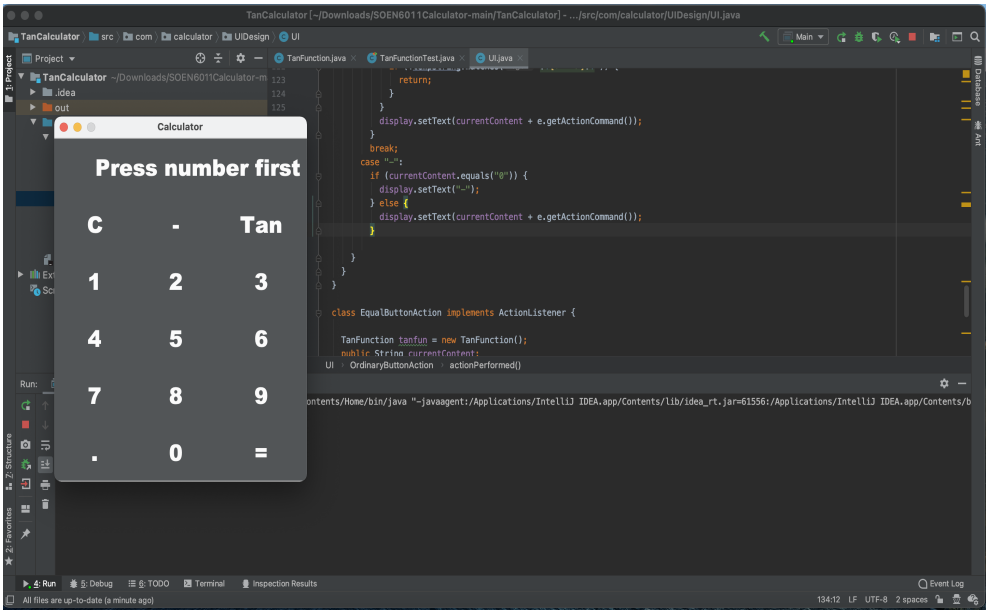
If there is an invalid input, the system will catch an exception and return an error message. After receive an error message, users can reenter the valid input and get the result they want.

Example:

1.User click "Tan" button first then click number



2. After click "=" button, return an error message. And use can use "C" button to clear the text and reenter the number according to the error message.



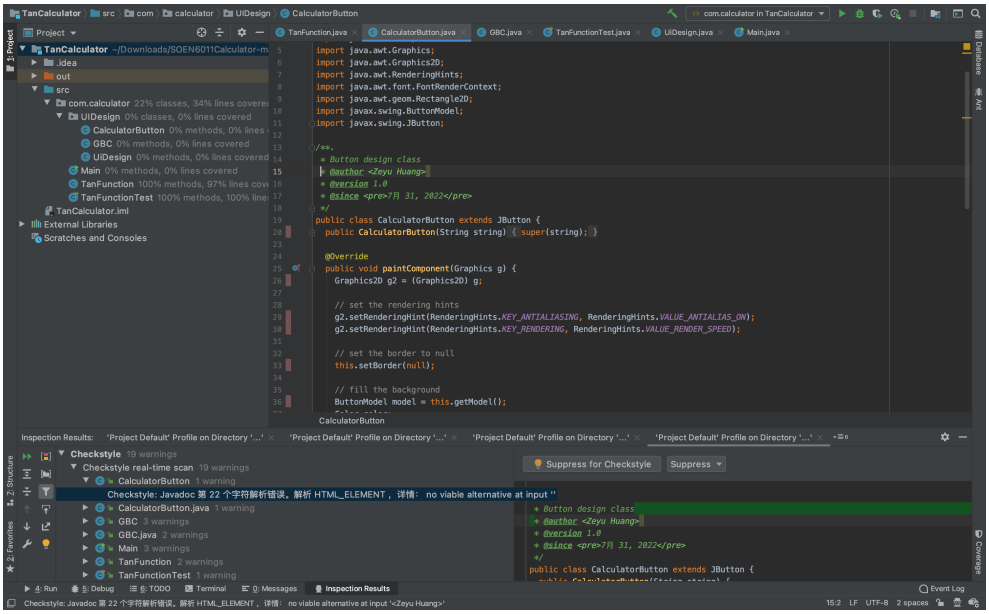
This is the code to handle this error.

```
@Override
public void actionPerformed(ActionEvent e) {
  currentContent = display.getText();

  currentContent = currentContent.substring(4, currentContent.length() - 1);
  System.out.println(currentContent + "!!!!!");
  try {
    display.setText(tanfun.calculateTan(Double.parseDouble(currentContent)));
  } catch (NumberFormatException ex) {
    display.setText("Press number first");
  }
}
```

c)Pragmatic Quality Checking Tool

For code quality check part, I use Checkstyle to check the code quality.[6]Checkstyle is a development tool to help programmers write Java code that adheres to a coding standard. It automates the process of checking Java code to spare humans of this boring (but important) task. This makes it ideal for projects that want to enforce a coding standard.



I use Checkstyle tool to fix some major warnings and errors, but left some unknown and unimportant warnings still not fixed.

5)Problem 5

Test Cases:

id	TC-01
Trace to requirement	FR1
Description	To calculate the value of Tan(0)
Precondition	The calculator is already on.
Expected result	0
Steps	1.Enter 0 2. Press "Tan" button 3. Press "=" button 4. Return 0 as output

id	TC-02
Trace to requirement	FR2
Description	To calculate the value of Tan(195)
Precondition	The calculator is already on.
Expected result	0.2679
Steps	1.Enter 195 2. Press "Tan" button 3. Press "=" button 4. Return 0.2679 as output

id	TC-03
Trace to requirement	FR3
Description	To calculate the value of Tan(-10)
Precondition	The calculator is already on.
Expected result	-0.1763
Steps	1.Enter -10 2. Press "Tan" button 3. Press "=" button 4. Return -0.1763 as output

id	TC-04
Trace to requirement	FR4
Description	To calculate the value of Tan(90)
Precondition	The calculator is already on.
Expected result	invalid
Steps	1.Enter 90 2. Press "Tan" button 3. Press "=" button 4. Return "invalid" as output

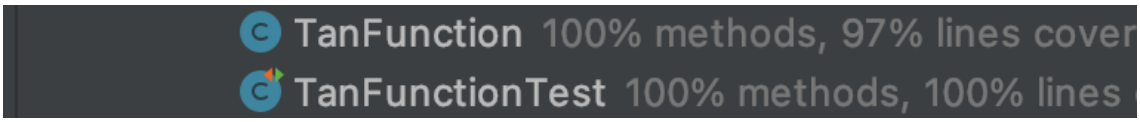
id	TC-05
Trace to requirement	FR1
Description	To calculate the value of Tan(180)
Precondition	The calculator is already on.
Expected result	-0
Steps	1.Enter 180 2. Press "Tan" button 3. Press "=" button 4. Return -0 as output

id	TC-06
Trace to requirement	FR2
Description	To calculate the value of Tan(45)
Precondition	The calculator is already on.
Expected result	1
Steps	1.Enter 45 2. Press "Tan" button 3. Press "=" button 4. Return 1 as output

id	TC-07
Trace to requirement	FR2
Description	To calculate the value of Tan(85)
Precondition	The calculator is already on.
Expected result	11.4300
Steps	1.Enter 85 2. Press "Tan" button 3. Press "=" button 4. Return 11.430 as output

id	TC-08
Trace to requirement	FR2
Description	To calculate the value of Tan(110)
Precondition	The calculator is already on.
Expected result	-2.7475
Steps	1.Enter 110 2. Press "Tan" button 3. Press "=" button 4. Return -2.7475 as output

For the test part, I use Junit testing framework to implement it. Below are the examples coverage and tests result.



Bibliography

[1] Varsity Tutors.
https://www.varsitytutors.com/hotmath/hotmath_help/topics/tangent-function

[2] varsitytutors.graphing tangent function.
https://www.varsitytutors.com/hotmath/hotmath_help/topics/graphing-tangent-function

[3] Polynomial approximation
https://www.expii.com/t/what-is-a-polynomial-approximation-317

[4] Maclaurin series
https://mathworld.wolfram.com/MaclaurinSeries.html

[5] Pseudocode for Polynomial approximation
https://mathonweb.com/help_ebook/html/algorithms.htm

[6] Checkstyle
https://checkstyle.sourceforge.io/

[7] Juit
https://en.wikipedia.org/wiki/JUnit