PostFix Calculator

Software Requirements Specification

Version 2.0

Team 2

Chauhan, Yash
Jin, Hui
Khan, Akbar
Martintsov, Vladimir
Nguyen, Andrew
Podimov, Michael

Prepared for
EECS 2311 - Software Development Project
Instructor: Gunnar Gotshalks
Winter 2016

PostFix Calculator

1. Introduction

This section of the document gives a description of the scope and an overview of the project. Also, the purpose and the functionality of the project is provided.

1.1 Purpose

The purpose of this document is to give a detailed description of the project "PostFix Calculator". This documentation is meant for Prof Gunnar Gotshalks and addresses purpose, constraints and development of the system. The intended audience for this document is anyone who intends to make a postfix graphing calculator.

1.2 Scope

"PostFix Calculator" is a Java based application that performs arithmetic, and trigonometric functions in postfix format. As a result of the calculator being postfix, the program implements a stack.
The software implements all the specifications.
The "PostFix Calculator" is scaled to a mobile device but is only deployed on a pc environment. The software is deployed through the use of Java Virtual Machine (JVM) in Java Archive(jar) format, available for download from our GitHub-based project website (http://vladimir95.github.io). This program can be deployed through Eclipse or terminal.

1.3 References

This document was adapted from a sample document, available at the following link

http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs_example_2010_group2.pdf

Specific requirements in the document are taken from the project requirements on the course web page.

1.4 Overview

The remainder of the document contains three chapters. First one talking about the overall description, functions, constraints, assumptions. The second one talks about the required features in detail from the customer and the third one has acceptance test cases for the various functions.

2. General Description

This section provides a brief overview of the system. This section covers the functionality of the system and shows the interface of the system. As well as an explanation of how the user interacts with the system.

2.1 Product Perspective

The “PostFix Calculator”, as previously mentioned, works in a postfix format also known as the reverse polish notation. This means that the system expects the operands before expecting the operator.
Therefore, the system accepts input from the user through the keys on the calculator. After each numeric input, the enter key needs to pressed to mark the ending of the operand and move on the the next operator/operand or calculate.

2.2 Product Functions

The “PostFix Calculator” is a very intuitive and user friendly calculator, as it does not require any parentheses. The calculator performs a wide range of functions, as provided in the table below.

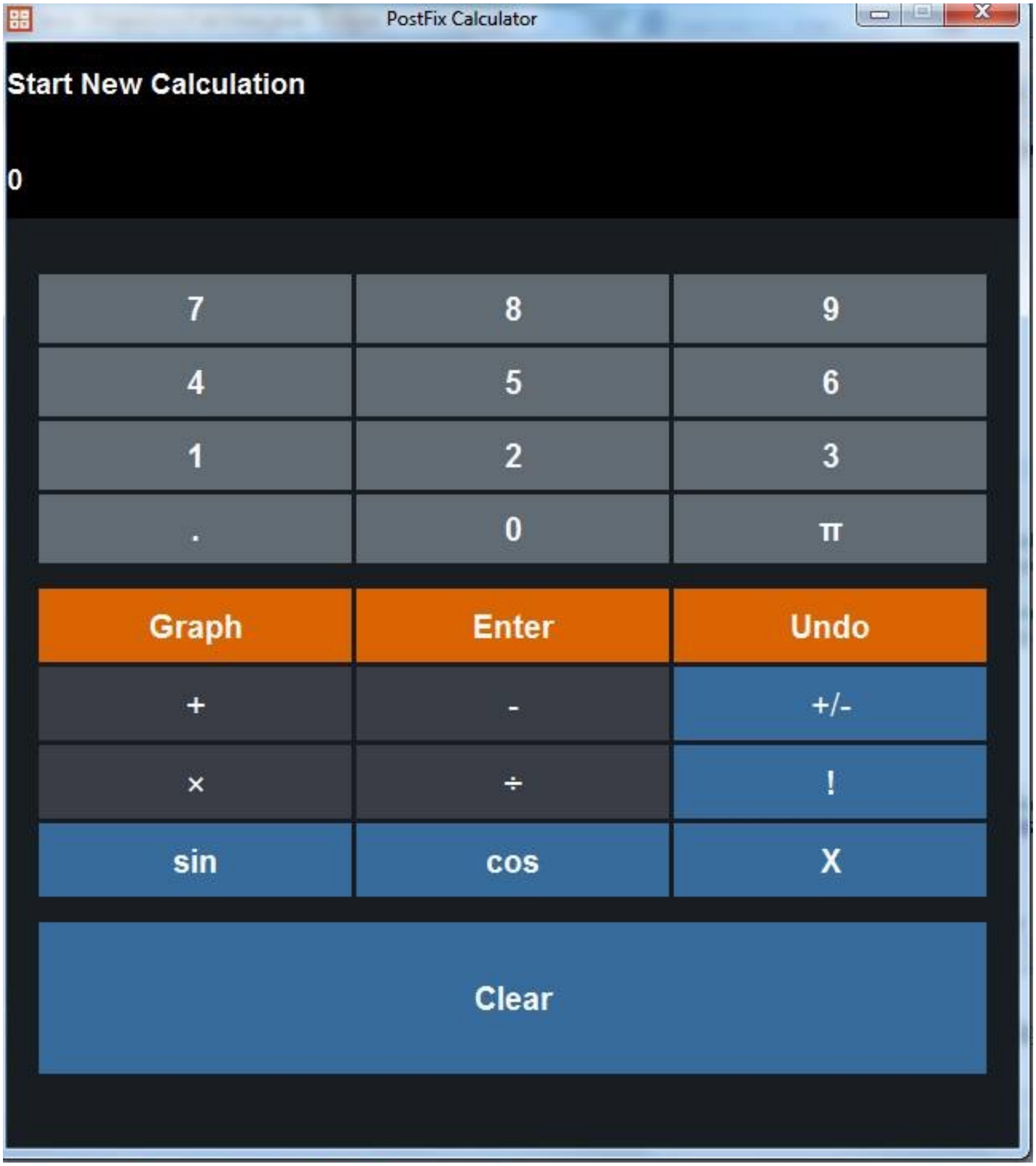
Key	Function
Main View	
0 - 9	Inputs the key pressed from zero to nine.
.	Inputs a decimal point. If there is no number prior to the decimal key, it is treated as if there is a 0 ahead of it.
π	Inputs pi to the top of the stack.
+	Performs the addition operation on the top two operands in the stack, and inputs the result on the top of the stack
-	Performs the subtraction operation on the top two operands in the stack, and inputs the result on the top of the stack
×	Performs the multiplication operation on the top two operands in the stack, and inputs the result on the top of the stack
÷	Performs the division operation on the top two operands in the stack, and inputs the result on the top of the stack
sin	Performs the sine function on the top operand in the stack, and inputs the result on the top of the stack
cos	Performs the cosine function on the top operand in the stack, and inputs the result on the top of the stack

Key	Function
Main View	
!	Performs the factorial operation on the top operand in the stack, , and inputs the result on the top of the stack
+ / -	Changes the sign of the input if the user is typing input, otherwise changes the sign of the top expression in the stack
Enter	Pushes the input to the top of the stack.
Undo	Undoes the previous action. If the user is typing input, deletes the last digit or decimal point.
Clear	Clears the input field and the memory of the calculator, therefore loses all the previous actions.
Graph	Loads a new view where the value in the stack is plotted. If the stack is empty, it is treated as 0.
X	Acts as a variable for expressions.
Graph View	
Back	Exits the graph view and goes back to the calculator view.
Favourites	Loads a view where a list of functions chosen as favourite appear.
Add to Favourites	Adds the current function on the graph to the list of favourites
Favourites View	
Back	Loads a view with the graph and previously plotted function appears.
Graph	Plots the function adjacent to the button on the graph view.
Delete	Removes the favourite function from the list of favourites.

An image of the interface is on the following page.

2.3 User Characteristics

The user for this software is any individual looking to calculate arithmetic functions in postfix or plot graphs. This software is developed for mobile devices but deployed on PC with JVM and can be accessed through Eclipse or launching it through terminal. The calculator takes input via the keys being pressed on the interface using the mouse. Input through the keyboard is not accepted.



Start New Calculation

0

2.4 General Constraints

The following design choices do not interfere in any way with the integrity of the software and meeting the specific requirements.
A constraint in the system, although mathematically correct, is that any division by 0 gives a Math Error.
Another constraint is that the factorial function will only operate on whole numbers and give a Math Error otherwise

3 Specific Requirements

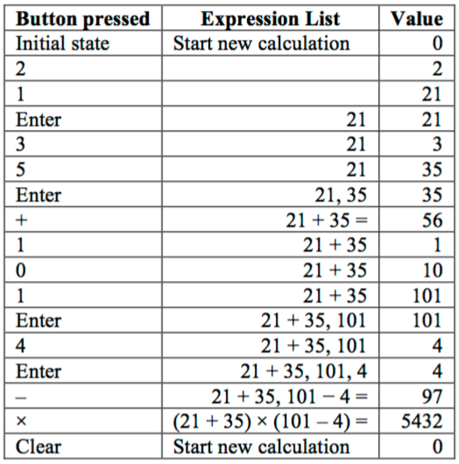
This section covers all the customer requirements that were used to guide the project's software design, implementation and the testing.

3.1 Interface Requirements

- The client wants the expression to be in infix notation, with a minimum number if parentheses.
  For example:
- Entry if numbers is in floating point; as real numbers. The output may be an integer, real number or in scientific notation; i.e., do the minimum work to display the result.
- A new button, "X" needs to be introduced as a variable in an expression.
- A graph label that plots a graph of the top expression in the expression stack.
- Different views, Calculator, graph and favourites, that you can switch between.

3.2 Functional Requirements

The system is required to have the following buttons and their functions that are described as well. This is similar to the product functions since all the requirements were implemented accordingly.



Button pressed	Expression List	Value
Initial state	Start new calculation	0
2		2
1		21
Enter	21	21
3	21	3
5	21	35
Enter	21, 35	35
+	21 + 35 =	56
1	21 + 35	1
0	21 + 35	10
1	21 + 35	101
Enter	21 + 35, 101	101
4	21 + 35, 101	4
Enter	21 + 35, 101, 4	4
–	21 + 35, 101 – 4 =	97
×	(21 + 35) × (101 – 4) =	5432
Clear	Start new calculation	0

Table 1: Calculating the expression (21 + 35) × (101 – 4).

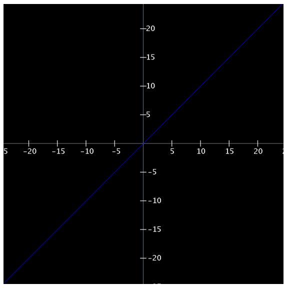
Key	Function
Main View	
0 - 9	Inputs the key pressed from zero to nine.
.	Inputs a decimal point. If there is no number prior to the decimal key, it is treated as if there is a 0 ahead of it.
π	This is a zero operand function and should be treated as a constant.
+	Performs the addition operation on the top two operands in the stack, and inputs the result on the top of the stack
-	Performs the subtraction operation on the top two operands in the stack, and inputs the result on the top of the stack
×	Performs the multiplication operation on the top two operands in the stack, and inputs the result on the top of the stack
÷	Performs the division operation on the top two operands in the stack, and inputs the result on the top of the stack
sin	Performs the sine function on the top operand in the stack, and inputs the result on the top of the stack
cos	Performs the cosine function on the top operand in the stack, and inputs the result on the top of the stack
!	Performs the factorial operation on the top operand in the stack, , and inputs the result on the top of the stack
+ / -	Changes the sign of the number in the value field. Function should be able to perform correctly without the use of ‘Enter’ key. If the user is in the middle of entering a number, the value should be changed and then let the user continue entering the number
Enter	Pushes the input to the top of the stack. If the user is typing input, deletes the last digit or decimal point.
Undo	While the user is typing in a number and ‘Undo’ is pressed, it should take back the last digit or decimal point pressed until doing so would clear the value entirely. If so, it should show the result of running the current expression in the expression list. If user is not typing and ‘Undo’ is pressed, it should remove the top item from the history list and update the display.
Clear	Clears the input field and the memory of the calculator, therefore loses all the previous actions.
Graph	Plots the expression on the top of the expression stack on the graph view.
X	Acts as a variable for expressions.
Graph View	
Back	Exits the graph view and goes back to the calculator view.
Favourites	Loads a view where a list of functions chosen as favourite appear.

Add to Favourites	Adds the current function on the graph to the list of favourites
Favourites View	
Back	Loads a view with the graph and previously plotted function appears.
Graph	Plots the function adjacent to the button on the graph view.
Delete	Removes the favourite function from the list of favourites.

4 Acceptance Cases

The following are test cases with results provided in the table. This should guide the explicit outputs and functions of all the features.
The commas represent that the value has been pushed in. (Enter key is pressed)

Feature	Expression	Output
Key input	9	9
Decimal	5.0	5.0
π	3π	3 x $\pi = 9.4247\ldots$
Addition	5, 5, +	5 + 5 = 10
Subtraction	9, 4, -	9 - 4 = 5
Multiplication	5, 6, x	5 x 6 = 30
Division	14, 7, ÷	14 ÷ 7 = 2
Sine Function	0, sin	sin(0) = 0
Cosine Function	π, cos	-1
Factorial Function	7, !	7! = 5040
Change Sign	6, + / -	-6 = -6
Enter	67, 9, 0,	67, 9, 0
Undo	While user is typing - 6689 Undo When user is not typing - 6, 5, + Undo	668 6, 5
Clear	67, 9, 87, 0, π, +, -, ! Clear	0

Feature	Expression	Output
Graph	X, Graph	
Back	Pressed while on Graph/Favourite view	Goes back to Calculator/Graph View
Favourites	When the favourites button is pressed	The favourites view appears
Add to Favourites	X, Graph, Add to Favourites	The function Y = X is added to the list of favourite functions and appears in Favourites view
Graph (While in favourites view)	Graph	The favourited function adjacent to the Graph button will be plotted on the Graph view.