Based on IEEE Std 830-1998.
Done by: Dmitrii Machurak
Date: 10/26/2023

Software Requirements Specification (SRS) for Algebraic Expression Evaluator

Table of Contents

1. Introduction

1.1 Purpose

- The purpose of this document is to specify the software requirements for an algebraic expression evaluator and its design. This software is designed to evaluate mathematical expressions, including basic arithmetic operations, functions, and variables.

1.2 Scope

- The algebraic expression evaluator is a Java-based library capable of evaluating mathematical expressions that include arithmetic operations and functions. It aims to provide a solution for developers needing to embed algebraic expression evaluation into their applications for various reasons

1.3 Definitions, Acronyms, and Abbreviations

- Algebraic Expression Evaluator (AEE)

1.4 References

- IEEE Recommended Practice for Software Requirements Specifications

2. Overall Description

2.1 Product Perspective

- The algebraic expression evaluator is a standalone Java library that relies only on built-in java math library.

2.2 Product Functions

- The software shall provide the following functionality:

    - **Expression Evaluation**: Evaluate algebraic expressions with support for addition, subtraction, multiplication, division, as well as, mathematical and trigonometric functions.

    - **Error Handling**: The AEE handles errors and provides clear error messages for incorrect expressions or syntax errors.

2.3 User Characteristics

- The intended users of this software are developers, who wish to embed algebraic expression evaluation into their applications, and students who wish to use the algebraic expression evaluator for educational purposes. Users should have a basic understanding of Java programming and algebraic expressions.

2.4 Constraints

- The software is constrained to the Java programming language and should run on Java-compatible platforms.

- Mathematical functions are based on Java's built-in Math library.

- Syntax and semantics for the AEE input is based on mathematical rules.

2.5 Assumptions and Dependencies

- The software assumes that valid algebraic expressions will be provided as input.
- The input is expected to be syntactically and semantically correct.
- It depends on the Java runtime environment.

3. Specific Requirements

3.1 Requirements

3.1.1 Functional Requirements

3.1.1.1 **Expression Evaluation**

- The software evaluates algebraic expressions that consist of numbers, arithmetic operations, mathematical functions.
- The software uses built-in methods to calculate trigonometric functions, power, and logarithm in base10.

3.1.1.2 **Expression Parsing**

- The software parses through the input using Recursive Descent Parser.
- It is critical that input has correct syntax to successfully parse the expression.

3.1.1.3 **Error Handling**

- The software provides clear and informative error messages for various errors, including syntactical and semantical errors, and undefined functions.
- Error messages aid developers and other users in debugging their expressions.

3.1.2 User Interaction

3.1.2.1 **Installation**

- To install the software, user is expected to download the “calc” directory from Git and read README file for further instructions.
- User is expected to compile the software ontheir machine either through command line or IDE.

3.1.2.2 **The use**

- After successful installation, user is expected to run the program using the folloing command “java calc/a”
- After the launch, user can enter the expression or type “exit” to close the program.

3.1.3 Non-Functional Requirements

3.1.3.1 **Performance**

- The software executes common algebraic expressions within a reasonable time frame, usually in milliseconds for moderately complex expressions.
- Response times should not exceed user expectations.

3.1.3.2 **Precision**

- The software shall provide reasonable precision for numerical calculations, considering the use of BigDecimal class that provides a high-level precision. Better than float or double representations of the numbers.
- Precision should be sufficient for most scientific and engineering applications.

3.1.3.3 **Usability**

- The software is easy to use and install. Besides that it is simple to embed into Java applications, offering a user-friendly experience for developers.

3.1.4 Feasibility Analysis

3.1.4.1 **Technical Feasibility**

- The software relies on Java, a widely-used and well-established developing programming language.

3.1.4.2 **Operational Feasibility**

- The software is operationally feasible and can be integrated into various Java applications simply.

3.2 Consistency Check

- The software performs consistency checks on algebraic expressions to ensure that they follow the Syntax rules of algebra and are internally consistent. This includes checking for balanced parentheses, valid function names, correct operator order.

3.3 Validity Check

- The software validates the syntax and semantics of algebraic expressions to ensure they are valid and meaningful. Including checking for proper function calls and semantical logic.

3.4 Prioritization of the Requirements

- The following requirements are prioritized based on their importance: Error handling, precision, usability, expression evaluation, expression parsing.

4. High-Level Design

5. Low-Level Design

5.1 Expression Parsing

5.2 Function Evaluation

- The software defines a set of recognized mathematical functions and evaluates them by using built-in methods.

5.3 Error Handling

- Software dynamically checks for errors at every stage of parsing and after error-prone functions like division and exponentiation.

6. Interfaces

6.1 API Interface

- The Java API interface provides methods for setting variables, evaluating expressions, and handling errors programmatically.

6.2 Console Interface

- A command-line interface implemented for manual testing and debugging purposes allowing user to paste expressions into the command line.