Hulk Programming Language Documentation

Francisco Prestamo

2023

1 Introduction

Hulk is a new programming language designed with simplicity and efficiency in mind. The language is currently in its early stages of development, and this documentation will provide an overview of its current capabilities.

2 Current Features

As of now, Hulk supports the following features:

- Lexing and parsing of numerical operations
- Execution of parsed operations to obtain a result

2.1 Numerical Operations

Hulk can process basic numerical operations involving addition, subtraction, multiplication, and division. The language can handle integer and floating-point numbers, as well as simple arithmetic expressions.

2.2 Code Explanation

2.2.1 Parsing

The user input lines, which are numerical expressions, these lines are read and processed. Once the input is obtained, the Parser.cs class is invoked. The Parser.cs class is responsible for parsing the input and generating the abstract syntactic tree. To accomplish this task, the Parser.cs class takes the help of the Lexer.cs class instances. The Lexer.cs class breaks down the input into tokens and feeds them to the Parser.cs class. These tokens are then analyzed by the Parser.cs class to identify the structure of the program. As the Parser.cs class processes the tokens, it simultaneously builds the abstract syntactic tree.

2.2.2 Evaluation

The Evaluator.cs is a crucial component of any compiler or interpreter. Once the abstract syntax tree (AST) has been generated by the parser, the Evaluator.cs takes over and evaluates the expressions represented by the tree. The Evaluator.cs works by recursively visiting each node of the AST, starting with the root node. As it visits each node, it performs the necessary calculations specified by the node's type and produces a final output. The Evaluator.cs continues to visit each child node of the current node until all nodes have been evaluated.

3 Example

Here's an example of a simple Hulk program that calculates the result of an arithmetic expression:

```
1 + 2 * 3 - 4 / 2
```

This program will output the result '5' after executing the operations in the correct order following the rules of precedence.

4 Future Development

The development of Hulk is ongoing, and the following features are planned for future releases:

- Support for variables and assignment
- Control structures (e.g., if-else statements, loops)
- Functions and modules
- Error handling and debugging tools

5 Conclusion

Hulk is a promising new programming language with a focus on simplicity and efficiency. Although the language is currently limited in its functionality, it provides a solid foundation for future development. As the language evolves, it is expected to gain more advanced features and become more versatile for a wider range of applications.