Dhanya Mary Jacob 1207643992

Nikhil Aravind 1207456168

Suhas Xavier 1207389153

Aaditya Maheshwari 1207856165

Compiler and Virtual Machine

For a Programming Language:

DANS

Team 1

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Serial No.** | **Topic** | **Page No.** |
| **1.** | **Introduction** | **3** |
| **2.** | **System Requirements** | **3** |
| **3.** | **Data Types** | **4** |
| **4.** | **Sample Programs** | **5** |
| **5.** | **References** | **8** |
|  |  |  |
|  |  |  |

1. **INTRODUCTION**

The aim of this project was to design, implement, and demonstrate our own language, compiler, and runtime for a language.

DANS is an ANTLR based implementation which allows a user to develop and compile basic programs using operators, primitive types, functions and loop constructs. DANS is entirely developed in ANTLR.

This project consists of a lexical analyzer to identify and isolate tokens specified by the language, a parser to parse the tokens and create a parse tree in order to understand what the user hopes to achieve, and an intermediate code generator which acts as the middle ground between the user’s high level language and the machine’s low level machine language. This intermediate code is then converted to machine code at runtime, and run in order to complete the task. This documentation covers how the user can install, run and program in DANS.

1. **SYSTEM REQUIREMENTS**

DANS has been developed using ANTLR in Eclipse Luna with ANTLR IDE. The link to download the environment can be found at the following link:

<https://marketplace.eclipse.org/content/antlr-ide>

The DANS programming language can be run by downloading the attached .jar file.

General System Requirements apply. We recommend having at least 500GB Hard Drive, 5 GB RAM with Windows XP, Vista, 7 or 8 Operating System. This is release 1.1 of the DANS programming language hence any system faults or issues faced can be mailed to any of the authors, and will be fixed in the next version.

1. DATA TYPES

DANS grammar is designed to work for the following constructs:

1. Assignment Statements
2. Function Calls
3. If Loops
4. While Loops
5. Expressions that allow for the following logical and relational operations:
6. Unary Minus
7. Not Expression
8. Multiply
9. Divide
10. Modulus
11. Add
12. Subtract
13. Greater than
14. Less than
15. Greater than equal to
16. Less than equal to
17. Equality
18. Not Equal
19. Logical Or
20. Logical And
21. Ternary Expression
22. Keywords used are:
23. Println
24. Input
25. Size
26. Def
27. If
28. Else
29. Else if
30. Return
31. For
32. While
33. To
34. Do
35. End
36. In
37. Null
38. Boolean Values:
39. True
40. False
41. Numbers, Integers and Digits
42. Identifiers and Strings
43. Comments
44. White Spaces

For a detailed explanation of what the grammar looks like and how each non terminal reaches a terminal in order to reach a derivation, consult the grammar file.

1. **SAMPLE PROGRAMS**

Using the following samples, you should be able to understand how DANS works and how to develop simple programs using DANS. For each example, a screen shot has been shown to describe what the code should look like, sample input and expected output.

* 1. – A simple Hello World Program

**println("Hello, World!");**

* 1. – An addition program

**a = 1;**

**b = 7;**

**sum = a + b;**

**product = a \* b;**

**println(sum);**

* 1. – A program using simple if conditional

**a = 4;**

**b = 8;**

**if a > b do**

**println("A is greater");**

**else do**

**println("B is greater");**

**end**

* 1. – While loop - factorial program

**number = 7;**

**fact = 1;**

**i = 1;**

**while i<=number do**

**fact = fact \* i;**

**i = i + 1;**

**end**

**println("The factorial of "+ number +" is: " + fact);**

1. **REFERENCES**

<http://www.antlr.org/>

<http://www.cse.chalmers.se/edu/course/DAT150/lectures/proglang-04.html>

<http://en.wikipedia.org/wiki/Lexical_analysis>

<http://www.antlr2.org/doc/lexer.html>