NOAH PROGRAMMING LANGUAGE

SER502 Team 15 | Notation for Optimized Algorithmic Handling

Arizona State University

TEAM MEMBERS

1. Om Rajesh Chauhan (1233682894)

2. Aksh Rajesh Chauhan (1233618037)

3. Harsh Sanjay Gohel (1233613227)

4. Nisarg Hemal Desai (1233606714)

PROJECT OVERVIEW

NOAH (Notation for Optimized Algorithmic Handling) is a programming language designed to blend robust static typing with the flexibility of modern paradigms. This project, developed as part of the SER502 course, demonstrates the implementation of a high-performance, compiled language with advanced constructs for real-world use cases.

PROJECT OBJECTIVES

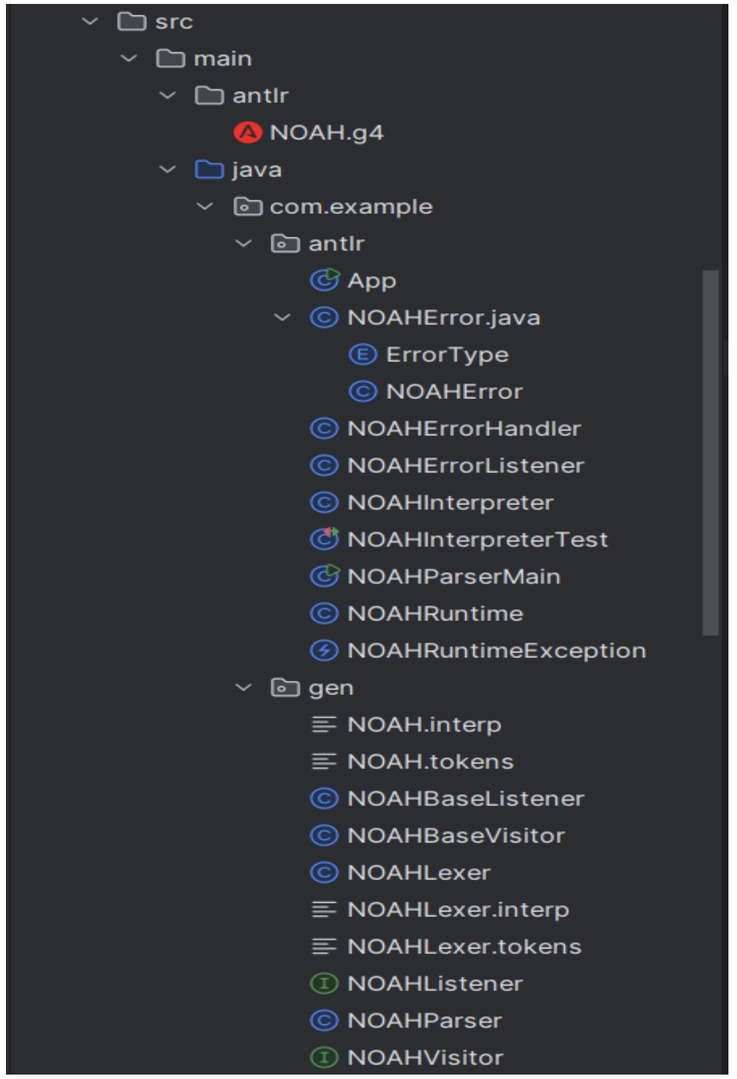
- Design a statically typed, high-performance language.
- Implement a robust parser and interpreter using ANTLR. Provide modern programming constructs for flexibility and usability.
- Demonstrate key features with well-documented sample programs.
- Ensure cross-platform compatibility and ease of use.

PROJECT ARCHITECTURE

- Components:
- 1. Lexical Analyzer
- Token generation
- Pattern matching
- Error detection

- 2. Parser
- Abstract Syntax Tree creation
- Grammar rule implementation
- Syntax validation

- 3. Interpreter/Runtime Environment
- Code execution
- Memory management
- Output handling



CORE LANGUAGE FEATURES

Data Types:*
- Boolean (true/false)
- Numeric (int, float)
- String with operations

Operators:*
- Arithmetic: +, -, *, /
- Boolean: and, or, not
- Relational: <, >, ==, !=

Control Structures:*
- Conditional statements
- Loop constructs
- Basic I/O operations

- Statically typed with compile-time type checking.

- Supports modern constructs such as conditional statements, loops, and operators.

- It Includes robust error handling mechanisms.

- Provides clear syntax and readability for developers.

GRAMMAR IMPLEMENTATION AND TOKEN DEFINITIONS

```
grammar NOAH;
program: statement* EOF;
statement: assignment
         | ifStatement
         | forLoop
         | whileLoop
         | printStatement;
```

Features:
- EBNF notation
- Clear syntactic rules
- Comprehensive expression handling

Token Types:*
- Keywords: if, else, for, while, print, true, false

- Operators: +, -, *, /, <, >, ==, !=, and, or, not, ?, :
- Delimiters: (,), {, }, ;, =

- Identifiers: [a-zA-Z_][a-zA-Z0-9_]*

- Numeric literals: [0-9]+(\.[0-9]+)?
- String literals: "[^"]*"

PARSER AND INTERPRETER

- The **NOAHParserMain** module converts NOAH code into an Abstract Syntax Tree (AST) and provides detailed error messages for syntax issues, ensuring code validity.

- The **NOAHInterpreter** processes the AST, executing nodes sequentially while managing variable declarations, evaluating expressions, and implementing control structures for program execution.

SAMPLE PROGRAMS

Program Types:-

- variables.noah: Variable operations
- operators.noah: Operator demonstrations
- booleans.noah: Logical operations
- control_flow.noah: Control structures
- loops.noah: Iteration examples
- logical1.noah: Nested conditionals

RUN INSTRUCTIONS

How to Run NOAH Programs

Steps:-

1. Navigate to project root

2. Execute command:Bash

```
java -cp target/classes com.example.antlr.NOAHParserMain
data/sample.noah
```

SYSTEM SUPPORT AND DEVELOPMENT TOOLS

Supported Operating Systems:

- Windows 10 and above
- macOS 10.15 (Catalina) and above
- Linux (Ubuntu 20.04 LTS and above)

Development Tools:

- ANTLR (4.13.2): Lexer and parser generation.
- GitHub: Version control and CI/CD pipelines.

FUTURE DEVELOPMENT

Planned Features:-

- Extended type system
- Advanced error recovery
- Performance optimization
- Additional language constructs
- Enhanced testing framework

CONCLUSION

Key Achievements:-

- Successful implementation of a new programming language
- Robust grammar and parser development
- Comprehensive testing and documentation
- Efficient team collaboration

TEAM CONTRIBUTIONS

Aksh Rajesh Chauhan:-
- Grammar rule definition
- Regular expression development
- Lexer optimization

Om Rajesh Chauhan:-
- ANTLR configuration
- Parser integration
- Error handling implementation

Nisarg Hemalkumar Desai:-
- Test program development
- Parser verification
- Designed sample NOAH programs

Harsh Sanjay Gohel:-
- Developed Interpreter
- Assisted in Testing & Integration
- Documentation & Repository management

THANK YOU

Arizona State University