Ministry of Education of Republic of Moldova

Technical University of Moldova

Faculty of Computers, Informatics and Microelectronics

Software Engineering Department

Computer Programming

Individual work

DSL - Calculator

Author:
Maxim Isacescu
std. gr. FAF-231

Verified:
Alexandru Furdui

Chișinău 2023

Theory Background

A domain-specific language (DSL) is a computer language specialized to a particular application domain. This is in contrast to a general-purpose language (GPL), which is broadly applicable across domains. DSLs can be further subdivided by the kind of language, and include domain-specific markup languages, domain-specific modeling languages, and domain-specific programming languages. Simpler DSLs, particularly ones used by a single application, are sometimes informally called mini-languages.

The Task

Create your own simple calculator Domain-Specific Language (DSL), similar to the example provided during our class. You have the freedom to make choices in the following aspects

Documentation

Input: File
Output: Terminal
Variables naming: only letters
Each statement must begin with new line
Math expressions are made from operators +, -, /, * and variables, integers, float numbers
Each element from the math expression must be separated by a space
Comments start with # and free lines are acceptable
Equal sign have two roles: assign values for new or existing variables; check equality of two values in if statements

Existing instructions:
<int/float><varname>= <math expression>
<varname>= <math expression>
show <math expression>

if <varname>= <math expression>then
...
endif

repeat <math expression>times
...
endrepeat

Possible errors:
Syntax Error: There is an if statement that is not 'closed' (unpredictable behavior may occur)"
Syntax Error: To many endif statements
Syntax Error: (on line number ...)
Division Error: You cannot divide by zero
Initialization Error: Cannot initiate one variable two times
Initialization Error: Variable was not initiated

Some errors and/or unpredictable behavior can be triggered if:
Math operator do not have 2 values to work with
Each element from the math expression is not separated by a space

Technical implementation

```
Open the file
Read line by line
Tokenize each line
Make AST of each tokenized line
Start parsing:
    Initiate arrays for variables and theirs values
    If it's␣if-statement␣-␣change␣condition␣state
␣␣␣␣If␣it's loop - start construct LoopList
    If condition state is active - statement starts to
       execute
    If loop condition is active - statement go in LoopList
       construction
    When the LoopList is closed - it's␣start␣execution
```

Conclusion

Creating a Domain-Specific Language (DSL) can significantly enhance productivity by tailoring a language to address specific issues within a well-defined domain. This targeted approach fosters collaboration and readability, allowing both developers and domain experts to work more effectively. However, developing a DSL requires careful planning, encompassing the definition of syntax, semantics, and potentially the creation of a compiler or interpreter.

Bibliography

1. https://github.com/M1XaM/DSL
2. https://en.wikipedia.org/wiki/Domain-specific-language
3. https://chat.openai.com/