МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Домашняя работа №4

Выполнил:

Круглов Егор Ильич, Р3324

Преподаватель:

Лаздин Артур Вячеславович

Санкт-Петербург 2025

Выполнение:

Код также выложен на github: https://github.com/KruglovEgor/Compilers/tree/main/lab4

Грамматика в формате g4:

```
program: statement+ EOF;
statement:
   | ifStatement
   | whileLoop
   | printStmt
   | inputStmt;
ifStatement: 'if' '(' expr ')' block ('else' block)?;
whileLoop: 'while' '(' expr ')' block;
printStmt: 'print' '(' expr ')' ';';
inputStmt: 'input' '(' ID ')' ';';
expr:
   expr op=('*' | '/' | '%') expr # MulDivMod
   | expr op=('+' | '-') expr
   | '(' expr ')'
| '-' expr
| '!' expr
WS: [ \t \n \r] + -> skip;
```

Интерпретатор:

```
from antlr4 import *
from MyLangLexer import MyLangLexer
from MyLangParser import MyLangParser
from MyLangVisitor import MyLangVisitor

class InterpreterError(Exception):
    pass
```

```
class MyInterpreter(MyLangVisitor):
   def visitProgram(self, ctx: MyLangParser.ProgramContext):
       for stmt in ctx.statement():
           self.visit(stmt)
   def visitAssignment(self, ctx: MyLangParser.AssignmentContext):
       var name = ctx.ID().getText()
       value = self.visit(ctx.expr())
       return value
   def visitPrintStmt(self, ctx: MyLangParser.PrintStmtContext):
   def visitInputStmt(self, ctx: MyLangParser.InputStmtContext):
   def visitIfStatement(self, ctx: MyLangParser.IfStatementContext):
       if self._is_true(self.visit(ctx.expr())):
           self.visit(ctx.block(0))
           self.visit(ctx.block(1))
   def visitWhileLoop(self, ctx: MyLangParser.WhileLoopContext):
           self.visit(ctx.block())
   def visitMulDivMod(self, ctx: MyLangParser.MulDivModContext):
       left = self.visit(ctx.expr(0))
       right = self.visit(ctx.expr(1))
               raise InterpreterError ("Деление по модулю на ноль")
           return left % right
   def visitAddSub(self, ctx: MyLangParser.AddSubContext):
       left = self.visit(ctx.expr(0))
       return left + right if op == '+' else left - right
   def visitCompare(self, ctx: MyLangParser.CompareContext):
```

```
left = self.visit(ctx.expr(0))
       right = self.visit(ctx.expr(1))
       if type(left) != type(right):
        }[op] else 0
   def visitString(self, ctx: MyLangParser.StringContext):
   def visitVariable(self, ctx: MyLangParser.VariableContext):
        var name = ctx.ID().getText()
   def visitParens(self, ctx: MyLangParser.ParensContext):
       return self.visit(ctx.expr())
   def visitUnaryMinus(self, ctx: MyLangParser.UnaryMinusContext):
   def visitLogicalNot(self, ctx: MyLangParser.LogicalNotContext):
def run interpreter(input stream):
   lexer = MyLangLexer(input stream)
   parser = MyLangParser(stream)
   tree = parser.program()
   interpreter = MyInterpreter()
       interpreter.visit(tree)
   except InterpreterError as e:
   if len(sys.argv) != 2:
   input stream = FileStream(sys.argv[1], encoding='utf-8')
   run interpreter(input stream)
```

Тесты:

Нахождение числа Фибонначи:

```
print("Введите номер числа Фибоначчи:");
input(i);

if (i < 0) {
    print("Число должно быть неотрицательным");
}
else{
    a = 0;
    b = 1;
    c = 0;

    if (i == 1) {
        c = 1;
    }

while (i - 1 > 0) {
        c = a + b;
        a = b;
        b = c;
        i = i - 1;
    }

print("Ваше число Фибоначчи:");
print(c);
}
```

```
(venv) PS E:\GitHub\Compilers\lab4> python my_interpreter.py tests/fibonachi.mylang
Введите номер числа Фибоначчи:
Ваше число Фибоначчи:
(venv) PS E:\GitHub\Compilers\lab4> python my_interpreter.py tests/fibonachi.mylang
Введите номер числа Фибоначчи:
1
Ваше число Фибоначчи:
1
(venv) PS E:\GitHub\Compilers\lab4> python my_interpreter.py tests/fibonachi.mylang
Введите номер числа Фибоначчи:
Ваше число Фибоначчи:
21
(venv) PS E:\GitHub\Compilers\lab4> python my_interpreter.py tests/fibonachi.mylang
Введите номер числа Фибоначчи:
Число должно быть неотрицательным
(venv) PS E:\GitHub\Compilers\lab4>
```

Нахождение НОД:

```
print("Нахождение НОД");
print("Введите первое число:");
input(a);
print("Введите второе число:");
input(b);

while (a * b != 0) {
    if (a > b) {
        a = a % b;
    }
    else{
        b = b % a;
    }
}
print("Получившийся НОД:");
print(a + b);
```

```
Нахождение НОД
Введите первое число:
63
Введите второе число:
49
Получившийся НОД:
(venv) PS E:\GitHub\Compilers\lab4> python my_interpreter.py tests/nod.mylang
Нахождение НОД
Введите первое число:
13
Введите второе число:
Получившийся НОД:
(venv) PS E:\GitHub\Compilers\lab4> python my_interpreter.py tests/nod.mylang
Нахождение НОД
Введите первое число:
50
Введите второе число:
125
Получившийся НОД:
25
```

Сложение чисел и строк:

```
a = 1;
b = 10;
c = "У нас ";
```

```
d = "получилось ";

print(a + b);

print(c + d);

print(c + d + a + b);

print(c + d + (a + b));
```

```
(venv) PS E:\GitHub\Compilers\lab4> python my_interpreter.py tests/sum.mylang
11
У нас получилось
У нас получилось 110
У нас получилось 11
```