

Отчет по заданию task 4 (task 10)

Ищенко Ю. А., 209 группа

Постановка задачи

Требуется разработать и реализовать интерпретатор модельного языка программирования. Инструментальный язык — C++. Модельный язык аналогичен языку программирования C.

Есть 3 типа данных: int, string, bool.

Операторы: if-else, read, write, while, do-while, оператор-выражение.

Операции: *, /, and, +, -, or, унарный +, унарный -, <, >, ==, <=, >=, !=, =.

Правила модельного языка:

- Объявление переменных происходит только в начале программы, при объявлении переменных может происходить присваивание им значений-констант.
- В любом месте программы, кроме идентификаторов, служебных слов и числовых констант, может находиться произвольное число пробельных литер и комментариев вида /* <любые символы, кроме */ > */.
- Пробел в строковой константе считается значащим символом строки.
- Внутри идентификаторов, служебных слов, числовых констант и разделителей, состоящих из нескольких символов, пробельные литеры недопустимы.

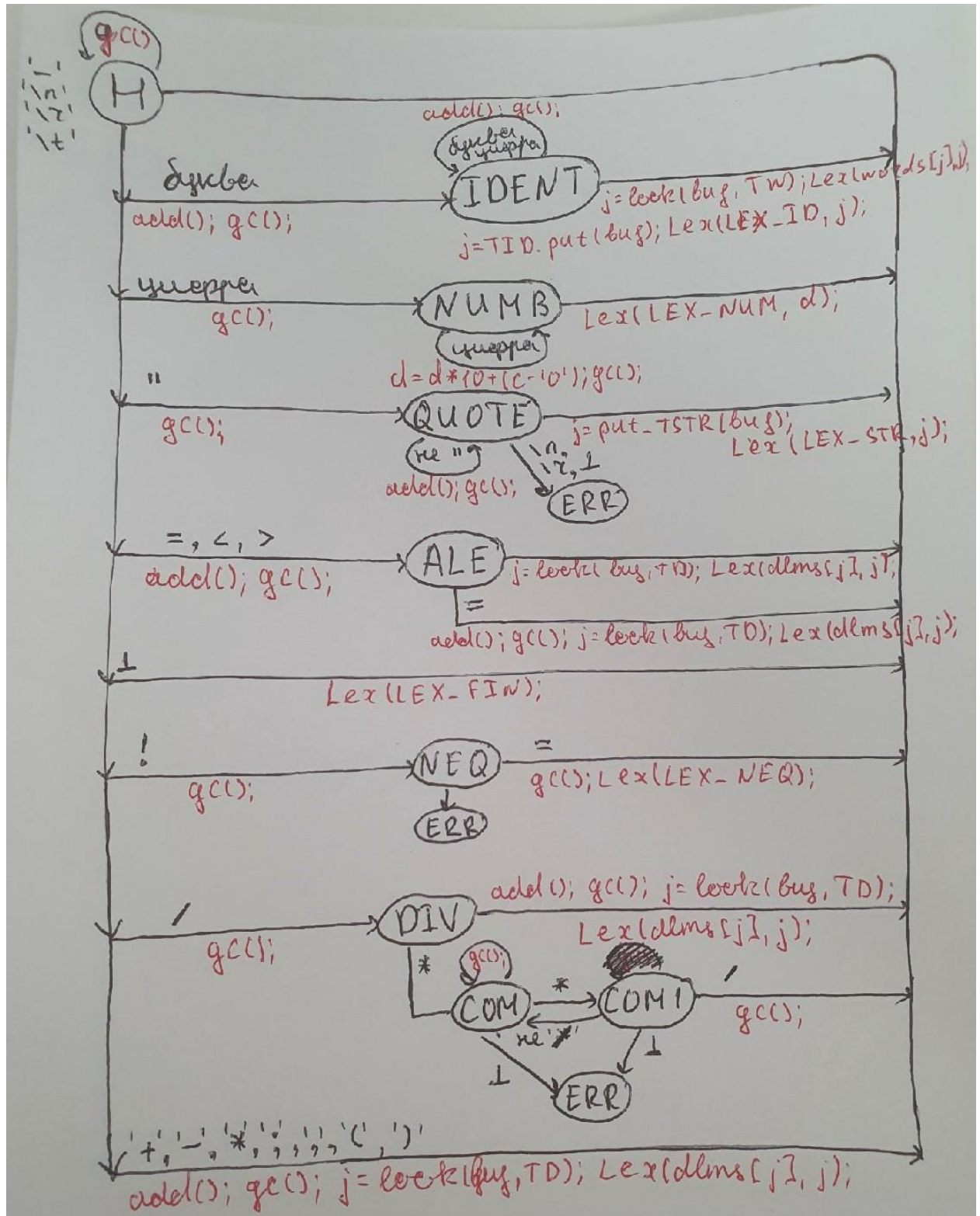
Синтаксис

$\langle \text{программа} \rangle \rightarrow \text{program } \{ \langle \text{описания} \rangle \langle \text{операторы} \rangle \}$
 $\langle \text{описания} \rangle \rightarrow \{ \langle \text{описание} \rangle ; \}$
 $\langle \text{описание} \rangle \rightarrow \langle \text{тип} \rangle \langle \text{переменная} \rangle \{ , \langle \text{переменная} \rangle \}$
 $\langle \text{тип} \rangle \rightarrow \text{int} \mid \text{string} \mid \text{bool}$
 $\langle \text{переменная} \rangle \rightarrow \langle \text{идентификатор} \rangle \mid \langle \text{идентификатор} \rangle = \langle \text{константа} \rangle$
 $\langle \text{константа} \rangle \rightarrow [+|-] \langle \text{целочисленная} \rangle \mid \langle \text{строковая} \rangle \mid \text{true} \mid \text{false}$
 $\langle \text{целочисленная} \rangle \rightarrow \langle \text{цифра} \rangle \{ \langle \text{цифра} \rangle \}$
 $\langle \text{строковая} \rangle \rightarrow " \{ \langle \text{литера} \rangle \} "$
 $\langle \text{операторы} \rangle \rightarrow \{ \langle \text{оператор} \rangle \}$
 $\langle \text{оператор} \rangle \rightarrow \text{if } (\langle \text{выражение} \rangle) \langle \text{оператор} \rangle \text{ else } \langle \text{оператор} \rangle \mid$
 $\text{while } (\langle \text{выражение} \rangle) \langle \text{оператор} \rangle \mid$
 $\text{read } (\langle \text{идентификатор} \rangle); \mid$
 $\text{write } (\langle \text{выражение} \rangle \{ , \langle \text{выражение} \rangle \}); \mid$
 $\text{do } \langle \text{оператор} \rangle \text{ while } (\langle \text{выражение} \rangle); \mid$
 $\langle \text{составной оператор} \rangle \mid \langle \text{оператор-выражение} \rangle$
 $\langle \text{составной оператор} \rangle \rightarrow \{ \langle \text{операторы} \rangle \}$
 $\langle \text{оператор-выражение} \rangle \rightarrow \langle \text{выражение} \rangle ;$
 $\langle \text{выражение} \rangle \rightarrow \langle \text{выражение 1 уровня} \rangle \mid \langle \text{идентификатор} \rangle = \langle \text{выражение} \rangle$
 $\langle \text{выражение 1 уровня} \rangle \rightarrow \langle \text{выражение 2 уровня} \rangle [< \mid > \mid == \mid <= \mid >= \mid !=]$
 $\langle \text{выражение 2 уровня} \rangle \mid \langle \text{выражение 2 уровня} \rangle$
 $\langle \text{выражение 2 уровня} \rangle \rightarrow [+ \mid -] \langle \text{выражение 3 уровня} \rangle$
 $\langle \text{выражение 3 уровня} \rangle \rightarrow \langle \text{выражение 4 уровня} \rangle \{ [+ \mid - \mid \text{or}] \langle \text{выражение 4 уровня} \rangle \}$
 $\langle \text{выражение 4 уровня} \rangle \rightarrow \langle \text{выражение 5 уровня} \rangle \{ [* \mid / \mid \text{and}] \langle \text{выражение 5 уровня} \rangle \}$
 $\langle \text{выражение 5 уровня} \rangle \rightarrow \langle \text{идентификатор} \rangle \mid \langle \text{строковая} \rangle \mid \text{true} \mid \text{false} \mid \text{not}$
 $\langle \text{выражение 5 уровня} \rangle \mid (\langle \text{выражение} \rangle)$

Составляющие программы

- Lex.h (класс лексем)
- Ident.h (класс идентификаторов)
- Scanner.h (класс для реализации лексического разбора: открытие файла, содержит таблицы TID (таблица идентификаторов), TSTR (таблица строк), объявление функции, реализующей лексический разбор)
- Scanner.cpp (функция, реализующая лексический разбор)
- Parser.h (класс, реализующий синтаксический, семантический разборы и перевод программы в ПОЛИЗ)
- Parser.cpp (реализация разборов)
- Executer.h (класс, реализующий выполнение программы на основе ПОЛИЗа)
- Executer.cpp (реализация функции выполнения)
- Interpretator.h (класс. в котором реализована функция вызова разбора программы и ее выполнения)
- main.cpp
- a_test\ (папка с тестами без ошибок)
- b_test\ (папка с тестами с ошибками)

Схема L-графа для реализации лексического разбора



Тесты

А-тесты:

```
program{
    int a = 51, b = 6, c=+1;
    string x = "abc 123", y, z = "abcd";
    bool d=false;
    int min=-5;
    write(min);
    c = (a + b) * 2;
    if ((c >= 100) or (x == z))
    {
        read (y);
        write (y);
        write (x + y + z, c);
    }
    else
        c = a = 21;
    a=-a;
    d=true or false;
    while( c > 100)
    {
        do{
            b=b+50;
        }
        while(b<200);
        c = c - 5;
        write (c);
        x = x + "step";
    }
    write (x);
    write("e123e");
}
```

```
program{
    1+1;
    write("Hello, world!", "Bye");
    while(false)
        write("false");
}
```

```
program {
    int x=2, y=5;
    x=(x+x+y)/2;
    write(x);
}
```

В-тесты:

```
program{
    int j=6, k;
    k=(i+j)*2; /* ERROR – нет описания переменной i */
    write("ERROR not detected");
}
```

```
program {
    string a = "abc";
    int b = 2, c;
    write(a + b); /* ERROR: different types */
}
```

```
program{
    int b;
    b=b+2; /* ERROR: b is not assigned */
}
```

```
program{
    int a=-2, b;
    b=a/0; /* ERROR: divide by zero */
}
```

```
program{
    int a=-2, b;
    string x="aaaa";
    b=a/x; /* ERROR: wrong types */
}
```