

Практикум на ЭВМ. Интерпретатор. Переменные

Баев А.Ж.

Казахстанский филиал МГУ

08 февраля 2020

Интерпретатор

1. Арифметические операторы
2. **Оператор присваивания**
3. Логические операторы
4. Оператор перехода (goto)
5. Условный оператор
6. Цикл while
7. Массивы
8. Функции
9. Рекурсия (стек для вызова функций)

Интерпретатор

1. Лексический анализ

строка

$x = 10 + y$

инфикс

| | | | | |
|---|---|----|---|---|
| x | = | 10 | + | y |
|---|---|----|---|---|

2. Синтаксический анализ

инфикс

| | | | | |
|---|---|----|---|---|
| x | = | 10 | + | y |
|---|---|----|---|---|

постфикс

| | | | | |
|---|----|---|---|---|
| x | 10 | + | y | = |
|---|----|---|---|---|

3. Вычисление

постфикс

| | | | | |
|---|----|---|---|---|
| x | 10 | y | + | = |
|---|----|---|---|---|

значение

12

Интерпретатор (переменные)

Пример:

$$x = 10 + y$$

Польская инверсная запись:

| | | | | |
|---|----|---|---|---|
| x | 10 | y | + | = |
|---|----|---|---|---|

Реализация классов

```
1  class Lexem {  
2  public:  
3      Lexem();  
4  };  
5  
6  class Number: public Lexem {  
7      int value;  
8  public:  
9      Number();  
10     int getValue();  
11 };
```

Реализация классов

```
1  enum OPERATOR {
2      ASSIGN,
3      PLUS, MINUS,
4      MULTIPLY,
5      LBRACKET, RBRACKET
6  };
7  int PRIORITY[] = {
8      0,
9      1, 1,
10     2,
11     3, 3
12 };
13 class Oper: public Lexem {
14     OPERATOR opertype;
15 public:
16     Oper();
17     OPERATOR getType();
18     int getPriority();
19     int getValue(const Number& left,
```

Реализация классов

Заведём класс переменных.

```
1  class Lexem {
2  public:
3      Lexem();
4  };
5
6  class Variable: public Lexem {
7      int value;
8      std::string name;
9  public:
10     Number();
11     int getValue();
12     void setValue(int value);
13 };
```

Реализация классов

Заведём таблицу переменных.

```
1 #include <map>
2
3 std::map<std::string, Variable*>
```


Особенность

Стек для вычислений теперь может содержать не только числа, но и переменные!

Левоассоциативность и правоассоциативность

Чем отличается порядок выполнения?

1 $x = y = 2$

1 $x + y + 2$

Левосторонность и правосторонность

1. Читаем очередной символ.
2. Если символ является числом, добавляем его к выходной строке.
3. Если символ является открывающей скобкой, помещаем его в стек.
4. Если символ является закрывающей скобкой, то выталкиваем все элементы из стека в выходную строку, пока верхним элементом стека не станет открывающая скобка.
5. Если символ является бинарной операцией «опер», то пока на вершине стека операция на вершине стека имеет приоритет больше (**или равен для левосторонней операции**), чем «опер», то выталкиваем верхний элемент стека в выходную строку, в конце помещаем «опер» в стек.
6. Когда входная строка закончилась, выталкиваем все символы из стека (операторы) в выходную строку.