**Описание языка SimpleCalc**

**Концепция:** Это простой язык для выполнения математических вычислений, работы с массивами и базовыми управляющими конструкциями.

**Особенности языка:**

1. **Типы данных:**
   * Целые числа (INT) и вещественные числа (FLOAT) — для переменных и констант;
   * Одномерные статические массивы (например, ID[S]);
   * Поддержка математических функций: sqrt, sin, cos.
2. **Операторы:**
   * Присваивание: =;
   * Арифметические операции: +, -, \*, / (приоритет: \*, / выше, чем +, -);
   * Операции сравнения: ==, !=, <, >, <=, >=;
   * Управляющие конструкции: if, else, while;
   * Ввод/вывод: input, print.
3. **Синтаксис:**
   * Скобки для группировки: (, ), [, ], {, };
   * Разделители: ;, ,.

**Список лексем**

Идентификаторы: ID (переменные, массивы);  
Константы: INT (целые числа), FLOAT (вещественные числа), STRING (строковые литералы);  
Операторы и знаки: = (присваивание), +, -, \*, /, ==, !=, <, >, <=, >=, - (унарный минус);  
Скобки: (, ), [, ], {, };  
Разделители: ;, ,;  
Служебные слова: if, else, while, input, print, sqrt, sin, cos.

**Таблица переходов автомата**

S0 – начальное состояние; S – конечное состояние; M – ошибочное состояние.

|  | **Буква** | **Цифра** | **+** | **-** | **\*** | **/** | **=** | **<** | **>** | **!** | **(** | **)** | **[** | **]** | **{** | **}** | **;** | **,** | **"** | **Др.** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S0 | S2/0 | S1/2 | S/8 | S/9 | S/7 | S/6 | S3/5 | S3/12 | S3/13 | S3/15 | S/10 | S/11 | S/20 | S/17 | S/27 | S/28 | S/26 | S/29 | S5/30 | M/19 |
| S1 | M/21 | S1/3 | S/16 | S/16 | S/16 | S/16 | S/16 | S/16 | S/16 | M/21 | S/16 | S/16 | S/16 | S/16 | S/16 | S/16 | S/16 | S/16 | M/21 | S/16 |
| S2 | S2/1 | S2/1 | S/15 | S/15 | S/15 | S/15 | S/15 | S/15 | S/15 | M/21 | S/15 | S/15 | S/15 | S/15 | S/15 | S/15 | S/15 | S/15 | M/21 | S/15 |
| S3 | M/21 | M/21 | M/21 | M/21 | M/21 | M/21 | S/14 | S/14 | S/14 | S/14 | M/21 | M/21 | M/21 | M/21 | M/21 | M/21 | M/21 | M/21 | M/21 | M/21 |
| S5 | S5/31 | S5/31 | S5/31 | S5/31 | S5/31 | S5/31 | S5/31 | S5/31 | S5/31 | S5/31 | S5/31 | S5/31 | S5/31 | S5/31 | S5/31 | S5/31 | S5/31 | S5/31 | S/32 | S5/31 |

**Примечание:**

* S3 используется для составных операторов (==, !=, <=, >=);
* S5 — для строковых констант.

**Семантические программы**

1. Начало идентификатора
2. Продолжение идентификатора
3. Начало константы
4. Продолжение константы
5. Распознано присваивание (=)
6. Распознано /
7. Распознано \*
8. Распознано +
9. Распознано -
10. Распознано (
11. Распознано )
12. Распознано <
13. Распознано >
14. Распознан составной оператор (==, !=, <=, >=)
15. Распознано имя
16. Распознана константа
17. Распознано ]
18. Не символ языка
19. Распознано [
20. Неверная лексема
21. Распознано ;
22. Распознано {
23. Распознано }
24. Распознано ,
25. Начало строки
26. Продолжение строки
27. Завершение строки

**КС-грамматика языка**

A – начальный нетерминал  
A → Q ; A | λ  
Q → E | C | L | I | O  
E → ID = S | ID [ S ] = S  
S → T U  
U → + T U | - T U | λ  
T → F W  
W → \* F W | / F W | λ  
F → ( S ) | ID | ID [ S ] | INT | FLOAT | - F | sqrt ( S ) | sin ( S ) | cos ( S )  
C → if ( V ) { A } B  
B → else { A } | λ  
L → while ( V ) { A }  
V → S K S  
K → == | != | < | > | <= | >=  
I → input ID  
O → print ( S ) | print ( STRING )

**КС-грамматика в нестрогой форме Грейбах**

A → Q ; A | λ  
Q → ID = S | ID [ S ] = S | if ( V ) { A } B | while ( V ) { A } | input ID | print ( S ) | print ( STRING )  
S → ( S ) U | ID U | ID [ S ] U | INT U | FLOAT U | - F U | sqrt ( S ) U | sin ( S ) U | cos ( S ) U  
U → + T U | - T U | λ  
T → ( S ) W | ID W | ID [ S ] W | INT W | FLOAT W | - F W | sqrt ( S ) W | sin ( S ) W | cos ( S ) W  
W → \* F W | / F W | λ  
B → else { A } | λ  
V → ( S ) K S | ID K S | ID [ S ] K S | INT K S | FLOAT K S | - F K S | sqrt ( S ) K S | sin ( S ) K S | cos ( S ) K S

**Семантические действия для генерации ОПС**

□ – пустое действие  
v – записать переменную в ОПС  
k – записать константу в ОПС  
+,-,\*,/,=,<,>,==,!=,<=,>= – записать операцию в ОПС  
i – индексация массива  
r – оператор чтения  
w – оператор вывода  
j – безусловный переход  
jf – переход если false  
m – создать метку

| **Правило** | **Действие** |
| --- | --- |
| A → Q ; A | Сгенерировать код для Q, затем для A |
| A → λ | □ |
| Q → E | Сгенерировать код для E |
| E → ID = S | Вычислить S, затем = v |
| E → ID [ S ] = S | Вычислить первый S (индекс), затем второй S, затем i = v |
| S → T U | Сгенерировать код для T, затем для U |
| U → + T U | Сгенерировать код для T, затем + и U |
| U → - T U | Сгенерировать код для T, затем - и U |
| U → λ | □ |
| T → F W | Сгенерировать код для F, затем для W |
| W → \* F W | Сгенерировать код для F, затем \* и W |
| W → / F W | Сгенерировать код для F, затем / и W |
| W → λ | □ |
| F → ID | v |
| F → ID [ S ] | Вычислить S, затем i v |
| F → INT | k |
| F → FLOAT | k |
| F → - F | Сгенерировать код для F, затем - |
| F → sqrt ( S ) | Вычислить S, затем sqrt |
| C → if ( V ) { A } B | m1, вычислить V, jf m2, сгенерировать A, j m3, m2, сгенерировать B, m3 |
| B → else { A } | Сгенерировать код для A |
| B → λ | □ |
| L → while ( V ) { A } | m1, вычислить V, jf m2, сгенерировать A, j m1, m2 |
| V → S K S | Вычислить первый S, затем второй S, затем K |
| I → input ID | r v |
| O → print ( S ) | Вычислить S, затем w |
| O → print ( STRING ) | k w |

**Список операций ОПС**

**Бинарные:**

* (сложение), - (вычитание), \* (умножение), / (деление), = (присваивание), i (индекс массива), < (меньше), > (больше), == (равно), != (не равно), <= (меньше или равно), >= (больше или равно)

**Унарные:**

* (унарный минус), r (чтение), w (вывод), sqrt (квадратный корень), sin (синус), cos (косинус)

**Переходы:**  
j (безусловный переход), jf (переход если false), m (метка)

**Пример ОПС**

Для выражения x = 5 + 3.14;:

k5 k3.14 + =x