## Операции языка (первая цифра варианта) представлены в таблицах 5.1 – 5.4.

Таблица 5.1 - Операции группы «отношение»

|  |  |
| --- | --- |
| Номер | Синтаксис группы операций (в порядке следования: неравно, равно, меньше, меньше или равно, больше, больше или равно) |
| 1 | <операции\_группы\_отношения>:: = < > | = | < | <= | > | >= |

Таблица 5.2 - Операции группы «сложение»

|  |  |
| --- | --- |
| Номер | Синтаксис группы операций (в порядке следования: сложение, вычитание, дизъюнкция) |
| 1 | <операции\_группы\_сложения>:: = + | - | *or* |

Таблица 5.3 - Операции группы «умножение»

|  |  |
| --- | --- |
| Номер | Синтаксис группы операций(в порядке следования: умножение, деление, конъюнкция) |
| 1 | <операции\_группы\_умножения>::= \* | / | *and* |

Таблица 5.4 - Унарная операция

|  |  |
| --- | --- |
| Номер | Синтаксис операции |
| 1 | <унарная\_операция>::= *not* |

Выражения языказадаются правилами:

<выражение>::= <операнд>{<операции\_группы\_отношения> <операнд>}

<операнд>::= <слагаемое> {<операции\_группы\_сложения> <слагаемое>}

<слагаемое>::= <множитель> {<операции\_группы\_умножения> <множитель>}

<множитель>::= <идентификатор> | <число> | <логическая\_константа> |

<унарная\_операция> <множитель> | (<выражение>)

<число>::= <целое> | <действительное>

<логическая\_константа>::= *true* | *false*

## Правила, определяющие идентификатор, букву и цифру:

<идентификатор>::= <буква> {<буква> | <цифра>}

<буква>::= *A* | *B* | *C* | *D* | *E* | *F* | *G* | *H* | *I* | *J* | *K* | *L* | *M* | *N* | *O* | *P* | *Q* | *R* | *S* | *T* |

*U* | *V* | *W* | *X* | *Y* | *Z* | *a* | *b* | *c* | *d* | *e* | *f* | *g* | *h* | *i* | *j* | *k* | *l* | *m* | *n* | *o* | *p*

*q* | *r* | *s* | *t* | *u* | *v* | *w* | *x* | *y* | *z*

<цифра>::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Правила, определяющие целые числа:

<целое>::= <двоичное> | <восьмеричное> | <десятичное> |

<шестнадцатеричное>

<двоичное>::= {/ 0 | 1 /} (*B* | *b*)

<восьмеричное>::= {/ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 /} (*O* | *o*)

<десятичное>::= {/ <цифра> /} [*D* | *d*]

<шестнадцатеричное>::= <цифра> {<цифра> | *A* | *B* | *C* | *D* | *E* | *F* | *a* | *b* |

*c* | *d* | *e* | *f*} (*H* | *h*)

Правила, описывающие действительные числа:

<действительное>::= <числовая\_строка> <порядок> |

[<числовая\_строка>] . <числовая\_строка> [порядок]

<числовая\_строка>::= {/ <цифра> /}

<порядок>::= ( E | e )[+ | -] <числовая\_строка>

Правила, определяющие структуру программы (вторая цифра варианта), представлены в таблице 5.5.

Таблица 5.5 – Структура программы

|  |  |
| --- | --- |
| Номер | Структура программы |
| 3 | <программа> = {/ (<описание> | <оператор>) переход строки /} *end* |

Правила, определяющиераздел описания переменных (третья цифра варианта) показаны в таблице 5.6.

Таблица 5.6 - Синтаксис команд описания данных

|  |  |
| --- | --- |
| Номер | Синтаксис команд описания данных |
| 2 | <описание>::= *dim* <идентификатор> {, <идентификатор> } <тип> |

## Правила, определяющие типы данных (четвертая цифра варианта) представлены в таблице 5.7.

Таблица 5.7- Описание типов данных

|  |  |
| --- | --- |
| Номер | Описание типов  (в порядке следования: целый, действительный, логический) |
| 1 | <тип>::= % | ! | $ |

## Правило, определяющее оператор программы (пятая цифра варианта).

<оператор>::= <составной> | <присваивания> | <условный> |

<фиксированного\_цикла> | <условного\_цикла> | <ввода> | <вывода>

Составной операторописан в таблице 5.8.

Таблица 5.8 - Синтаксис составного оператора

|  |  |
| --- | --- |
| Номер | Синтаксис оператора |
| 1 | <составной>::= <оператор> { : <оператор> } |

Оператор присваиванияописан в таблице 5.9.

Таблица 5.9 - Синтаксис оператора присваивания

|  |  |
| --- | --- |
| Номер | Оператор присваивания |
| 1 | <присваивание>::= <идентификатор> *ass* <выражение> |

Оператор условного перехода задан в таблице 5.10.

Таблица 5.10 - Синтаксис оператора условного перехода

|  |  |
| --- | --- |
| Номер | Оператор условного перехода |
| 1 | <условный>::= *if* <выражение> *then* <оператор> [ *else* <оператор>] |

Оператор цикла с фиксированным числом повторений описан в таблице 5.11.

Таблица 5.11 - Синтаксис оператора цикла с фиксированным числом повторений

|  |  |
| --- | --- |
| Номер | Синтаксис оператора |
| 1 | <фиксированного\_цикла>::= *for* <присваивания> *to* <выражение> *do* <оператор> |

Условный оператор циклазадан в таблице 5.12.

Таблица 5.12 - Синтаксис условного оператора цикла

|  |  |
| --- | --- |
| Номер | Синтаксис оператора |
| 1 | <условного\_цикла>::= *while* <выражение> *do* <оператор> |

Оператор вводаописан в таблице 5.13.

Таблица 5.13 - Синтаксис оператора ввода

|  |  |
| --- | --- |
| Номер | Синтаксис оператора |
| 1 | <ввода>::= *read* (<идентификатор> {, <идентификатор> }) |

Оператор выводапредставлен в таблице 5.14.

Таблица 5.14 - Синтаксис оператора вывода

|  |  |
| --- | --- |
| Номер | Синтаксис оператора |
| 1 | <вывода>::= *write* (<выражение> {, <выражение> }) |

Многострочные комментарии в программе(шестая цифра варианта) определены в таблице 5.15. Индивидуальные номера вариантов представлены в таблице 5.16.

Таблица 5.15 – Синтаксис многострочных комментариев

|  |  |  |
| --- | --- | --- |
| Номер | Признак начала комментария | Признак конца комментария |
| 1 | \\* | \*\ |

Текст программы должен допускать использование комментариев.

Пример программы:

// Тестовая программа

/\*

Объявляем переменные

\*/

dim a, b, c %

// Присваиваем

a ass 111b

/\* Вводим \*/

read(b)

c ass a + b

if c > 20 then write(“Прикол”) else write(“Не прикол”)

end