

11 вариант 21112

1. Операции языка

1.1. Операции группы «отношение»

2) <операции_группы_отношения>::= != | = | < | <= | > | >=

1.2. Операции группы «сложение»

2) <операции_группы_сложения>::= + | - | ||

1.3. Операции группы «умножение»

2) <операции_группы_умножения>::= * | / | &&

1.4. Унарная операция

2) <унарная_операция>::= !

2. Правила, определяющие структуру программы

2.1. Структура программы

1) <программа>::= program var <описание> begin <оператор> {; <оператор>} end.

3. Правила, определяющие раздел описания переменных

3.1. Синтаксис команд описания данных

1) <описание>::= {<идентификатор> {, <идентификатор>} : <тип> ;}

4. Правила, определяющие типы данных

4.1. Описание типов данных

1) <тип>::= % | ! | \$

5. Правило, определяющее оператор программы

<оператор>::= <составной> | <присваивания> | <условный> | <фиксированного_цикла> | <условного_цикла> | <ввода> | <вывода>

5.1. Синтаксис составного оператора

2) <составной>::= begin <оператор> {; <оператор>} end

5.2. Синтаксис оператора присваивания

2) <присваивания>::= <идентификатор> := <выражение>

5.3. Синтаксис оператора условного перехода

2) <условный>::= if «(<выражение>)» <оператор> [else <оператор>]

5.4. Синтаксис оператора цикла с фиксированным числом повторений

2) <фиксированного_цикла>::= for <присваивания> to <выражение> [step <выражение>] <оператор> next

5.5. Синтаксис условного оператора цикла

2) <условного_цикла>::= while «(<выражение>)» <оператор>

5.6. Синтаксис оператора ввода

2) <ввода>::= readln идентификатор {, <идентификатор>}

5.7. Синтаксис оператора вывода

2) <вывода>::= writeln <выражение> {, <выражение>}

6. Многострочные комментарии в программе

1) { ... }

Правила языка для всех вариантов:

Выражения языка задаются правилами:

1. <выражение>::= <операнд>{<операции_группы_отношения> <операнд>}
2. <операнд>::= <слагаемое> {<операции_группы_сложения> <слагаемое>}
3. <слагаемое>::= <множитель> {<операции_группы_умножения> <множитель>}
4. <множитель>::= <идентификатор> | <число> | <логическая_константа> |
<унарная_операция> <множитель> | «(<выражение>)>
5. <число>::= <целое> | <действительное>
6. <логическая_константа>::= true | false

Правила, определяющие идентификатор, букву и цифру:

7. <идентификатор>::= <буква> {<буква> | <цифра>}
8. <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
9. <цифра>::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Правила, определяющие целые числа:

10. <целое>::= <двоичное> | <восьмеричное> | <десятичное> |
<шестнадцатеричное>
11. <двоичное>::= { / 0 | 1 / } (B | b)
12. <восьмеричное>::= { / 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 / } (O | o)
13. <десятичное>::= { / <цифра> / } [D | d]
14. <шестнадцатеричное>::= <цифра> {<цифра> | A | B | C | D | E | F | a | b | c | d |
e | f} (H | h)

Правила, описывающие действительные числа:

15. <действительное>::= <числовая_строка> <порядок> | [<числовая_строка>] .
<числовая_строка> [порядок]
16. <числовая_строка>::= { / <цифра> / }
17. <порядок>::= (E | e) [+ | -] <числовая_строка>

Напиши на питоне графическое приложение, использует модуль tkinter. Приложению в разрешении 1400x1050. Это эмуляция транслятора псевдо языка программирования. Также добавь в приложение возможность выбора цветовой темы: светлая или темная. По темам должно быть похоже на цвета Visual Studio Code. Программа имеет текстовое поле ввода текста(почти во весь экран), под ним аналог консоли вывода. Кнопку справа сверху запуска этого кода. Кнопку, сверху вывода информации, о том, какой синтаксис и правила у нашего языка программирования. Ты должен сделать шаблон этого приложения, но обработчики синтаксиса и грамматики псевдо языка я сам буду писать. Твоя задача - сделать приложение-каркас. Приложение по сути свое IDE.