ТЕМАТИКА КУРСОВЫХ РАБОТ

Каждый студент выбирает работу согласно номеру варианта. Каждому номеру варианта соответствует 5-значное число, каждая цифра в котором отвечает определенной синтаксической особенности модельного языка программирования

Разработать программно-математическую модель распознавателя языка программирования. Вариант №:

1. 11111	6. 12121	11. 21112	16. 22312	21. 31111	26. 32112
2. 12221	7. 12311	12. 21322	17. 22332	22. 31121	27. 32122
3. 11321	8. 12331	13. 21332	18. 23112	23. 31131	28. 32312
4. 11331	9. 13111	14. 22112	19. 23222	24. 33221	29. 33113
5. 12113	10. 13211	15. 22222	20. 23332	25. 31331	30. 33123

Номера заданий складываются по следующим правилам:

1. Операции языка (*первая цифра варианта*)

1.1. Операции группы «отношение»

1) <операции_группы_отношения>::= <> | = | < | <= | > | >=

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

3) <операции_группы_отношения>::= NE | EQ | LT | LE | GT | GE

1.2. Операции группы «сложение»

1) <операции_группы_сложения>::= + | - | or

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

3) <операции_группы_сложения>::= plus | min | or

1.3. Операции группы «умножение»

1) <операции_группы_умножения>::= * | / | and

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

3) <операции_группы_умножения>::= mult | div | and

1.4. Унарная операция

1) <унарная_операция>::= not

2) <унарная_операция>::= !

3) <унарная_операция>::= ~

2. Правила, определяющие структуру программы (*вторая цифра варианта*)

2.1. Структура программы

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

2) <программа>::= «{» {/ (<описание> | <оператор>) ; /} «}»

3) <программа> = {/ (<описание> | <оператор>) (: | переход строки) /} end

3. Правила, определяющие раздел описания переменных (*третья цифра варианта*)

3.1. Синтаксис команд описания данных

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

2) <описание>::= dim <идентификатор> {, <идентификатор> } <тип>

3) <описание>::= <тип> <идентификатор> { , <идентификатор> }

4. Правила, определяющие типы данных (*четвертая цифра варианта*)

4.1. Описание типов данных

1) <тип>::= % | ! | $

2) <тип>::= integer | real | boolean

3) <тип>::= int | float | bool

5. Правило, определяющее оператор программы (*пятая цифра варианта*)

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

5.1. Синтаксис составного оператора

1) <составной>::= «[» <оператор> { (: | перевод строки) <оператор> } «]»

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

3) <составной>::= «{» <оператор> { ; <оператор> } «}»

5.2. Синтаксис оператора присваивания

1) <присваивания>::= <идентификатор> as <выражение>

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

3) <присваивания>::= [let] <идентификатор> = <выражение>

5.3. Синтаксис оператора условного перехода

1) <условный>::= if <выражение> then <оператор> [else <оператор>]

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

3) <условный>::= if <выражение> then <оператор> [else <оператор>] end_else

5.4. Синтаксис оператора цикла с фиксированным числом повторений

1) <фиксированного_цикла>::= for <присваивания> to <выражение> do <оператор>

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

3) <фиксированного_цикла>::= for «(»[<выражение>] ; [<выражение>] ; [<выражение>] «)» <оператор>

5.5. Синтаксис условного оператора цикла

1) <условного_цикла>::= while <выражение> do <оператор>

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

3) <условного_цикла>::= do while <выражение> <оператор> loop

5.6. Синтаксис оператора ввода

1) <ввода>::= read «(»<идентификатор> {, <идентификатор> } «)»

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

3) <ввода>::= input «(»<идентификатор> {пробел <идентификатор>} «)»

5.7. Синтаксис оператора вывода

1) <вывода>::= write «(»<выражение> {, <выражение> } «)»

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

3) <вывода>::= output «(»<выражение> { пробел <выражение> } «)»