

Вариант 1

Function \rightarrow Begin Descriptions Operators End
Begin \rightarrow Type FunctionName() {
End \rightarrow return Id ; }
FunctionName \rightarrow Id
Descriptions \rightarrow Descr | Descr Descriptions
Operators \rightarrow Op | Op Operators
Descr \rightarrow Type VarList ;
VarList \rightarrow Id | Id VarList
Type \rightarrow int | float
Op \rightarrow Id = Expr ;
SimpleExpr \rightarrow Id | Const | (Expr) | itof (Expr) | ftoi (Expr)
Expr \rightarrow SimpleExpr | SimpleExpr + Expr | SimpleExpr - Expr
Id \rightarrow **id_name**
Const \rightarrow **int_num** | **float_num**

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число, **float_num** – слово, представляющее собой вещественное число (два целых числа, разделенных точкой), itof – функция преобразования целого числа в вещественное, itod – функция преобразования вещественного числа в целое.

Вариант 2

Function \rightarrow Begin Descriptions Operators End
Begin \rightarrow Type FunctionName() {
End \rightarrow return Id ; }
FunctionName \rightarrow Id
Descriptions \rightarrow Descr | Descr Descriptions
Operators \rightarrow Op | Op Operators
Descr \rightarrow Type VarList ;
VarList \rightarrow Id | Id , VarList
Type \rightarrow int | double
Op \rightarrow Id = Expr ;
SimpleExpr \rightarrow Id | Const | (Expr) | itod (Expr) | dtod (Expr)
Expr \rightarrow SimpleExpr | SimpleExpr + Expr | SimpleExpr - Expr
Id \rightarrow **id_name**
Const \rightarrow **int_num** | **double_num**

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, а также цифр (цифры должны идти только после букв), **int_num** – слово, представляющее собой целое число, **double_num** – слово, представляющее собой вещественное число (два целых числа, разделенных точкой), itod – функция преобразования целого числа в вещественное, dtod – функция преобразования вещественного числа в целое.

Вариант 3

Function \rightarrow Begin Descriptions Operators End
Begin \rightarrow Type FunctionName() {
End \rightarrow return Id ; }
FunctionName \rightarrow Id
Descriptions \rightarrow Descr | Descr Descriptions
Operators \rightarrow Op | Op Operators
Descr \rightarrow Type VarList ;
VarList \rightarrow Id | Id , VarList
Type \rightarrow int | double
Op \rightarrow Id = Expr ;
SimpleExpr \rightarrow Id | Const | (Expr)
Expr \rightarrow Term | Term + Expr | Term - Expr
Term \rightarrow SimpleExpr | SimpleExpr * Term | SimpleExpr / Term
Id \rightarrow **id_name**
Const \rightarrow **int_num** | **double_num**

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, а также цифр (цифры должны идти только после букв), **int_num** – слово, представляющее собой целое число, **double_num** – слово, представляющее собой вещественное число (два целых числа, разделенных точкой).

Вариант 4

Function \rightarrow Begin Descriptions Operators End
Begin \rightarrow Type FunctionName() {
End \rightarrow return Id ; }
FunctionName \rightarrow Id
Descriptions \rightarrow Descr | Descr Descriptions
Operators \rightarrow Op | Op Operators
Descr \rightarrow Type VarList ;
VarList \rightarrow Id | Id , VarList
Type \rightarrow int | float
Op \rightarrow Id = Expr ;
SimpleExpr \rightarrow Id | Const | (Expr)
Expr \rightarrow Term | Term + Expr | Term - Expr
Term \rightarrow SimpleExpr | SimpleExpr * Term | SimpleExpr / Term | SimpleExpr % Term
Id \rightarrow **id_name**
Const \rightarrow **int_num** | **double_num**

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число, **double_num** – слово, представляющее собой вещественное число (два целых числа, разделенных точкой).

Вариант 5

Function \rightarrow Begin Descriptions Operators End
Begin \rightarrow Type FunctionName() {
End \rightarrow return Id ; }
FunctionName \rightarrow Id
Descriptions \rightarrow Descr | Descr Descriptions
Operators \rightarrow Op | Op Operators
Descr \rightarrow Type VarList ;
VarList \rightarrow Id | Id , VarList
Type \rightarrow int | char
Op \rightarrow Id = NumExpr ; | Id = StringExpr ;
SimpleNumExpr \rightarrow Id | Const | (NumExpr)
NumExpr \rightarrow SimpleNumExpr | SimpleNumExpr + NumExpr | SimpleNumExpr - NumExpr
SimpleStringExpr \rightarrow **char _const**
StringExpr \rightarrow SimpleStringExpr | SimpleStringExpr + StringExpr
Id \rightarrow **id _name**
Const \rightarrow **int _num**

Здесь **id _name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int _num** – слово, представляющее собой целое число, **char _const** – строка, составленная из заглавных и строчных букв латинского алфавита, а также цифр, выделенная двойными кавычками.

Вариант 6

Function \rightarrow Begin Descriptions Operators End
Begin \rightarrow Type FunctionName() {
End \rightarrow return Id ; }
FunctionName \rightarrow Id
Descriptions \rightarrow Descr | Descr Descriptions
Operators \rightarrow Op | Op Operators
Descr \rightarrow Type VarList ;
VarList \rightarrow Id | Id , VarList
Type \rightarrow int
Op \rightarrow Id = Expr ; | if (Condition) { Operators } else { Operators }
SimpleExpr \rightarrow Id | Const | (Expr) |
Expr \rightarrow SimpleExpr | SimpleExpr + Expr | SimpleExpr - Expr
Condition \rightarrow Expr RelationOperators Expr
RelationOperators \rightarrow == | != | < | > | <= | >=
Id \rightarrow **id _name**
Const \rightarrow **int _num**

Здесь **id _name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int _num** – слово, представляющее собой целое число.

Вариант 7

Function \rightarrow Begin Descriptions Operators End
Begin \rightarrow Type FunctionName() {
End \rightarrow return Id ; }
FunctionName \rightarrow Id
Descriptions \rightarrow Descr | Descr Descriptions
Operators \rightarrow Op | Op Operators
Descr \rightarrow Type VarList ;
VarList \rightarrow Id | Id , VarList
Type \rightarrow int
Op \rightarrow Id = Expr ; | while (Condition) { Operators }
SimpleExpr \rightarrow Id | Const | (Expr) Expr \rightarrow SimpleExpr | SimpleExpr + Expr | SimpleExpr - Expr
Condition \rightarrow Expr RelationOperators Expr
RelationOperators \rightarrow == | != | < | > | <= | >=
Id \rightarrow **id_name**
Const \rightarrow **int_num**
Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число.

Вариант 8

Function \rightarrow Begin Descriptions Operators End
Begin \rightarrow Type FunctionName() {
End \rightarrow return Id ; }
FunctionName \rightarrow Id
Descriptions \rightarrow Descr | Descr Descriptions
Operators \rightarrow Op | Op Operators
Descr \rightarrow Type VarList ;
VarList \rightarrow Id | Id , VarList
Type \rightarrow int
Op \rightarrow Id = Expr ; | do { Operators } while (Condition)
SimpleExpr \rightarrow Id | Const | (Expr)
Expr \rightarrow SimpleExpr | SimpleExpr + Expr | SimpleExpr - Expr
Condition \rightarrow Expr RelationOperators Expr
RelationOperators \rightarrow == | != | < | > | <= | >=
Id \rightarrow **id_name**
Const \rightarrow **int_num**
Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число.

Вариант 9

Function \rightarrow Begin Descriptions Operators End

Begin \rightarrow Type FunctionName() {
 End \rightarrow return Id ; }
 FunctionName \rightarrow Id
 Descriptions \rightarrow Descr | Descr Descriptions
 Operators \rightarrow Op | Op Operators
 Descr \rightarrow Type VarList ;
 VarList \rightarrow Id | Id , VarList
 Type \rightarrow int
 Op \rightarrow Id = Expr ; | for (Id = Expr; Condition ; Expr) { Operators }
 SimpleExpr \rightarrow Id | Const | (Expr) |
 Expr \rightarrow SimpleExpr | SimpleExpr + Expr | SimpleExpr - Expr
 Condition \rightarrow Expr RelationOperators Expr
 RelationOperators \rightarrow == | != | < | > | <= | >=
 Id \rightarrow **id_name**
 Const \rightarrow **int_num**

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число.

Вариант 10

Function \rightarrow Begin Descriptions Operators End
 Begin \rightarrow Type FunctionName() {
 End \rightarrow return Id ; }
 FunctionName \rightarrow Id
 Descriptions \rightarrow Descr | Descr Descriptions
 Operators \rightarrow Op | Op Operators
 Descr \rightarrow Type VarList ;
 VarList \rightarrow Id | Id , VarList
 Type \rightarrow int
 Op \rightarrow Id = Expr ; | switch (Expr) { Options }
 Options \rightarrow case Const : Operators break; | case Const : Operators break; Options
 SimpleExpr \rightarrow Id | Const | (Expr) Expr \rightarrow SimpleExpr | SimpleExpr + Expr | SimpleExpr
 - Expr
 Id \rightarrow **id_name**
 Const \rightarrow **int_num**

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число.

Вариант 11

Function \rightarrow Begin Descriptions Operators End
 Begin \rightarrow Type FunctionName() {
 End \rightarrow return Id ; }
 FunctionName \rightarrow Id
 Descriptions \rightarrow Descr | Descr Descriptions

$\text{Operators} \rightarrow \text{Op} \mid \text{Op Operators}$
 $\text{Descr} \rightarrow \text{Type VarList} ;$
 $\text{VarList} \rightarrow \text{Id} \mid \text{Id} , \text{VarList}$
 $\text{Type} \rightarrow \text{int}$
 $\text{Op} \rightarrow \text{Id} = \text{Expr} ; \mid \text{goto Mark}; \mid \text{Mark} : \text{Operator}$
 $\text{Mark} \rightarrow \text{Id}$
 $\text{SimpleExpr} \rightarrow \text{Id} \mid \text{Const} \mid (\text{Expr})$
 $\text{Expr} \rightarrow \text{SimpleExpr} \mid \text{SimpleExpr} + \text{Expr} \mid \text{SimpleExpr} - \text{Expr}$
 $\text{Id} \rightarrow \text{id_name}$
 $\text{Const} \rightarrow \text{int_num}$

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число.

Вариант 12

$\text{Program} \rightarrow \text{Begin Descriptions Operators End}$
 $\text{Begin} \rightarrow \text{PROGRAM Id}$
 $\text{End} \rightarrow \text{END Id}$
 $\text{Descriptions} \rightarrow \text{Descr} \mid \text{Descr Descriptions}$
 $\text{Operators} \rightarrow \text{Op} \mid \text{Op Operators}$
 $\text{Descr} \rightarrow \text{Type VarList}$
 $\text{Type} \rightarrow \text{INTEGER} \mid \text{REAL}$
 $\text{VarList} \rightarrow \text{Id} \mid \text{Id} , \text{VarList}$
 $\text{Op} \rightarrow \text{Id} = \text{Expr}$
 $\text{SimpleExpr} \rightarrow \text{Id} \mid \text{Const} \mid (\text{Expr}) \mid \text{ITOR} (\text{Expr}) \mid \text{RTOI} (\text{Expr})$
 $\text{Expr} \rightarrow \text{SimpleExpr} \mid \text{SimpleExpr} + \text{Expr} \mid \text{SimpleExpr} - \text{Expr}$
 $\text{Id} \rightarrow \text{id_name}$
 $\text{Const} \rightarrow \text{int_num} \mid \text{real_num}$

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число, **real_num** – слово, представляющее собой вещественное число (два целых числа, разделенных точкой), **ITOR** – функция преобразования целого числа в вещественное, **ITOD** – функция преобразования вещественного числа в целое.

Вариант 13

$\text{Program} \rightarrow \text{Begin Descriptions Operators End}$
 $\text{Begin} \rightarrow \text{PROGRAM Id}$
 $\text{End} \rightarrow \text{END Id}$
 $\text{Descriptions} \rightarrow \text{Descr} \mid \text{Descr Descriptions}$
 $\text{Operators} \rightarrow \text{Op} \mid \text{Op Operators}$
 $\text{Descr} \rightarrow \text{Type VarList}$
 $\text{Type} \rightarrow \text{INTEGER} \mid \text{COMPLEX}$
 $\text{VarList} \rightarrow \text{Id} \mid \text{Id} , \text{VarList}$
 $\text{Op} \rightarrow \text{Id} = \text{Expr}$

$\text{SimpleExpr} \rightarrow \text{Id} \mid \text{Const} \mid (\text{Expr}) \mid \text{CMPLX} (\text{Expr} , \text{Expr}) \mid \text{CABS} (\text{Expr})$
 $\text{Expr} \rightarrow \text{SimpleExpr} \mid \text{SimpleExpr} + \text{Expr} \mid \text{SimpleExpr} - \text{Expr}$
 $\text{Id} \rightarrow \text{id_name}$
 $\text{Const} \rightarrow \text{int_num} \mid \text{complex_num}$

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число, **complex_num** – слово, представляющее пару из двух целых чисел, разделенных запятой и взятой в скобки. Например, (2, 3), CABS – функция, вычисляющая модуль комплексного числа, CMPLX – функция, преобразовывающая два выражения в комплексное число (первый аргумент в вещественную часть, второй – в мнимую).

Вариант 14

$\text{Program} \rightarrow \text{Begin Descriptions Operators End}$
 $\text{Begin} \rightarrow \text{PROGRAM Id}$
 $\text{End} \rightarrow \text{END Id}$
 $\text{Descriptions} \rightarrow \text{Descr} \mid \text{Descr Descriptions}$
 $\text{Operators} \rightarrow \text{Op} \mid \text{Op Operators}$
 $\text{Descr} \rightarrow \text{Type VarList}$
 $\text{Type} \rightarrow \text{INTEGER} \mid \text{CHARACTER}$
 $\text{VarList} \rightarrow \text{Id} \mid \text{Id} , \text{VarList}$
 $\text{Op} \rightarrow \text{Id} = \text{NumExpr} \mid \text{Id} = \text{StringExpr}$
 $\text{SimpleNumExpr} \rightarrow \text{Id} \mid \text{Const} \mid (\text{NumExpr})$
 $\text{NumExpr} \rightarrow \text{SimpleNumExpr} \mid \text{SimpleNumExpr} + \text{NumExpr} \mid \text{SimpleNumExpr} - \text{NumExpr}$
 $\text{SimpleStringExpr} \rightarrow \text{string_const}$
 $\text{StringExpr} \rightarrow \text{SimpleStringExpr} \mid \text{SimpleStringExpr} + \text{StringExpr}$
 $\text{Id} \rightarrow \text{id_name}$
 $\text{Const} \rightarrow \text{int_num}$

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число, **string_const** – строка, составленная из заглавных и строчных букв латинского алфавита, а также цифр, выделенная одинарными кавычками.

Вариант 15

$\text{Program} \rightarrow \text{Begin Descriptions Operators End}$
 $\text{Begin} \rightarrow \text{PROGRAM Id}$
 $\text{End} \rightarrow \text{END Id}$
 $\text{Descriptions} \rightarrow \text{Descr} \mid \text{Descr Descriptions}$
 $\text{Operators} \rightarrow \text{Op} \mid \text{Op Operators}$
 $\text{Descr} \rightarrow \text{Type VarList}$
 $\text{Type} \rightarrow \text{INTEGER}$
 $\text{VarList} \rightarrow \text{Id} \mid \text{Id} , \text{VarList}$
 $\text{Op} \rightarrow \text{Id} = \text{Expr} \mid \text{IF Condition THEN Operators} \mid \text{IF Condition THEN Operators ELSE Operators}$

$\text{SimpleExpr} \rightarrow \text{Id} \mid \text{Const} \mid (\text{Expr})$
 $\text{Expr} \rightarrow \text{SimpleExpr} \mid \text{SimpleExpr} + \text{Expr} \mid \text{SimpleExpr} - \text{Expr}$
 $\text{Condition} \rightarrow \text{Expr} \text{RelationOperators} \text{Expr}$
 $\text{RelationOperators} \rightarrow .\text{EQ.} \mid .\text{NE.} \mid .\text{GE.} \mid .\text{GT.} \mid .\text{LE.} \mid .\text{LT.}$
 $\text{Id} \rightarrow \text{id_name}$
 $\text{Const} \rightarrow \text{int_num}$

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число.

Вариант 16

$\text{Program} \rightarrow \text{Begin} \text{Descriptions} \text{Operators} \text{End}$
 $\text{Begin} \rightarrow \text{PROGRAM} \text{Id}$
 $\text{End} \rightarrow \text{END} \text{Id}$
 $\text{Descriptions} \rightarrow \text{Descr} \mid \text{Descr} \text{Descriptions}$
 $\text{Operators} \rightarrow \text{Op} \mid \text{Op} \text{Operators}$
 $\text{Descr} \rightarrow \text{Type} \text{VarList}$
 $\text{Type} \rightarrow \text{INTEGER} \mid \text{REAL}$
 $\text{VarList} \rightarrow \text{Id} \mid \text{Id} , \text{VarList}$
 $\text{Op} \rightarrow \text{Id} = \text{Expr} \mid \text{CALL} \text{Id} (\text{VarList})$
 $\text{SimpleExpr} \rightarrow \text{Id} \mid \text{Const} \mid (\text{Expr})$
 $\text{Expr} \rightarrow \text{SimpleExpr} \mid \text{SimpleExpr} + \text{Expr} \mid \text{SimpleExpr} - \text{Expr}$
 $\text{Id} \rightarrow \text{id_name}$
 $\text{Const} \rightarrow \text{int_num} \mid \text{real_num}$

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число, **real_num** – слово, представляющее собой вещественное число (два целых числа, разделенных точкой).

Вариант 17

$\text{Program} \rightarrow \text{Begin} \text{Descriptions} \text{Operators} \text{End}$
 $\text{Begin} \rightarrow \text{PROGRAM} \text{Id}$
 $\text{End} \rightarrow \text{END} \text{Id}$
 $\text{Descriptions} \rightarrow \text{Descr} \mid \text{Descr} \text{Descriptions}$
 $\text{Operators} \rightarrow \text{Op} \mid \text{Op} \text{Operators}$
 $\text{Descr} \rightarrow \text{Type} \text{VarList}$
 $\text{Type} \rightarrow \text{INTEGER}$
 $\text{VarList} \rightarrow \text{Id} \mid \text{Id} , \text{VarList}$
 $\text{Op} \rightarrow \text{Id} = \text{Expr} \mid \text{FOR} \text{Id} = \text{Expr} \text{ TO } \text{Expr} \text{ DO } \text{Operators}$
 $\text{SimpleExpr} \rightarrow \text{Id} \mid \text{Const} \mid (\text{Expr})$
 $\text{Expr} \rightarrow \text{SimpleExpr} \mid \text{SimpleExpr} + \text{Expr} \mid \text{SimpleExpr} - \text{Expr}$
 $\text{Id} \rightarrow \text{id_name}$
 $\text{Const} \rightarrow \text{int_num}$

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число.

Вариант 18

Procedure \rightarrow Begin Descriptions Operators End
Begin \rightarrow procedure ProcedureName ; begin
End \rightarrow end
ProcedureName \rightarrow Id
Descriptions \rightarrow var DescrList
DescrList \rightarrow Descr | Descr DescrList
Descr \rightarrow VarList : Type ;
VarList \rightarrow Id | Id , VarList
Type \rightarrow integer | real
Operators \rightarrow Op | Op Operators
Op \rightarrow Id := Expr ;
Expr \rightarrow SimpleExpr | SimpleExpr + Expr | SimpleExpr - Expr
SimpleExpr \rightarrow Id | Const | (Expr) | RTOI (Expr) | ITOR (Expr)
Id \rightarrow **id_name**
Const \rightarrow **int_num** | **real_num**

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число, **real_num** – слово, представляющее собой вещественное число (два целых числа, разделенных точкой).

Вариант 19

Procedure \rightarrow Begin Descriptions Operators End
Begin \rightarrow procedure ProcedureName ; begin
End \rightarrow end
ProcedureName \rightarrow Id
Descriptions \rightarrow VAR DescrList
DescrList \rightarrow Descr | Descr DescrList
Descr \rightarrow VarList : Type ;
VarList \rightarrow Id | Id , VarList
Type \rightarrow INTEGER | REAL
Operators \rightarrow Op | Op Operators
Op \rightarrow Id := Expr ;
Expr \rightarrow Term | Term + Expr | Term - Expr
Term \rightarrow SimpleExpr * Term | SimpleExpr div Term | SimpleExpr mod Term
SimpleExpr \rightarrow Id | Const | (Expr)
Id \rightarrow **id_name**
Const \rightarrow **int_num** | **real_num**

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число, **real_num** – слово, представляющее собой вещественное число (два целых числа, разделенных точкой).

Вариант 20

Procedure \rightarrow Begin Descriptions Operators End
Begin \rightarrow procedure ProcedureName ; begin
End \rightarrow end
ProcedureName \rightarrow Id
Descriptions \rightarrow var DescrList
DescrList \rightarrow Descr | Descr DescrList
Descr \rightarrow VarList : Type ;
VarList \rightarrow Id | Id , VarList
Type \rightarrow integer | char
Operators \rightarrow Op | Op Operators
Op \rightarrow Id := NumExpr ; | Id := StringExpr ;
NumExpr \rightarrow SimpleNumExpr | SimpleNumExpr + Expr | SimpleNumExpr - Expr
SimpleNumExpr \rightarrow Id | Const | (NumExpr)
StringExpr \rightarrow SimpleStringExpr | SimpleStringExpr + StringExpr
SimpleStringExpr \rightarrow **string_const**
Id \rightarrow **id_name**
Const \rightarrow **int_num**

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число, **string_const** – строка, составленная из заглавных и строчных букв латинского алфавита, а также цифр, выделенная двойными кавычками.

Вариант 21

Procedure \rightarrow Begin Descriptions Operators End
Begin \rightarrow procedure ProcedureName ; begin
End \rightarrow end
ProcedureName \rightarrow Id
Descriptions \rightarrow var DescrList
DescrList \rightarrow Descr | Descr DescrList
Descr \rightarrow VarList : Type ;
VarList \rightarrow Id | Id , VarList
Type \rightarrow integer
Operators \rightarrow Op | Op Operators
Op \rightarrow Id := Expr ; | if Condition then Op | if (Condition) then Op else Op
Expr \rightarrow SimpleExpr | SimpleExpr + Expr | SimpleExpr - Expr
SimpleExpr \rightarrow Id | Const | (Expr)
Condition \rightarrow Expr RelationOperator Expr
RelationOperator \rightarrow = | <> | > | <
Id \rightarrow **id_name**
Const \rightarrow **int_num**

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число.

Вариант 22

Procedure \rightarrow Begin Descriptions Operators End
Begin \rightarrow procedure ProcedureName ; begin
End \rightarrow end
ProcedureName \rightarrow Id
Descriptions \rightarrow var DescrList
DescrList \rightarrow Descr | Descr DescrList
Descr \rightarrow VarList : Type ;
VarList \rightarrow Id | Id , VarList
Type \rightarrow integer
Operators \rightarrow Op | Op Operators
Op \rightarrow Id := Expr ; | case Expr of Options end ;
Options \rightarrow Const : Operators | Const : Operators Options
Expr \rightarrow SimpleExpr | SimpleExpr + Expr | SimpleExpr - Expr
SimpleExpr \rightarrow Id | Const | (Expr)
Id \rightarrow **id_name**
Const \rightarrow **int_num**

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число.

Вариант 23

Procedure \rightarrow Begin Descriptions Operators End
Begin \rightarrow procedure ProcedureName ; begin
End \rightarrow end
ProcedureName \rightarrow Id
Descriptions \rightarrow var DescrList
DescrList \rightarrow Descr | Descr DescrList
Descr \rightarrow VarList : Type ;
VarList \rightarrow Id | Id , VarList
Type \rightarrow integer
Operators \rightarrow Op | Op Operators
Op \rightarrow Id := Expr ; | while Condition do begin Operators end ;
Expr \rightarrow SimpleExpr | SimpleExpr + Expr | SimpleExpr - Expr
SimpleExpr \rightarrow Id | Const | (Expr)
Condition \rightarrow Expr RelationOperator Expr
RelationOperator \rightarrow = | <> | > | <
Id \rightarrow **id_name**
Const \rightarrow **int_num**

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число.

Вариант 24

Procedure \rightarrow Begin Descriptions Operators End
Begin \rightarrow procedure ProcedureName ; begin
End \rightarrow end
ProcedureName \rightarrow Id
Descriptions \rightarrow var DescrList
DescrList \rightarrow Descr | Descr DescrList
Descr \rightarrow VarList : Type ;
VarList \rightarrow Id | Id , VarList
Type \rightarrow integer
Operators \rightarrow Op | Op Operators
Op \rightarrow Id := Expr ; | repeat Operators until Condition ;
Expr \rightarrow SimpleExpr | SimpleExpr + Expr | SimpleExpr - Expr
SimpleExpr \rightarrow Id | Const | (Expr)
Condition \rightarrow Expr RelationOperator Expr
RelationOperator \rightarrow = | <> | > | <
Id \rightarrow **id_name**
Const \rightarrow **int_num**

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число.

Вариант 25

Procedure \rightarrow Begin Descriptions Operators End
Begin \rightarrow procedure ProcedureName ; begin
End \rightarrow end
ProcedureName \rightarrow Id
Descriptions \rightarrow var DescrList
DescrList \rightarrow Descr | Descr DescrList
Descr \rightarrow VarList : Type ;
VarList \rightarrow Id | Id , VarList
Type \rightarrow integer
Operators \rightarrow Op | Op Operators
Op \rightarrow Id := Expr ; | for Id := Const to Const do begin Operators end ;
Expr \rightarrow SimpleExpr | SimpleExpr + Expr | SimpleExpr - Expr
SimpleExpr \rightarrow Id | Const | (Expr)
Condition \rightarrow Expr RelationOperator Expr
RelationOperator \rightarrow = | <> | > | <
Id \rightarrow **id_name**
Const \rightarrow **int_num**

Здесь **id_name** – слово, состоящее из заглавных и строчных букв латинского алфавита, **int_num** – слово, представляющее собой целое число.