Расширенные Формы Бэкуса – Наура

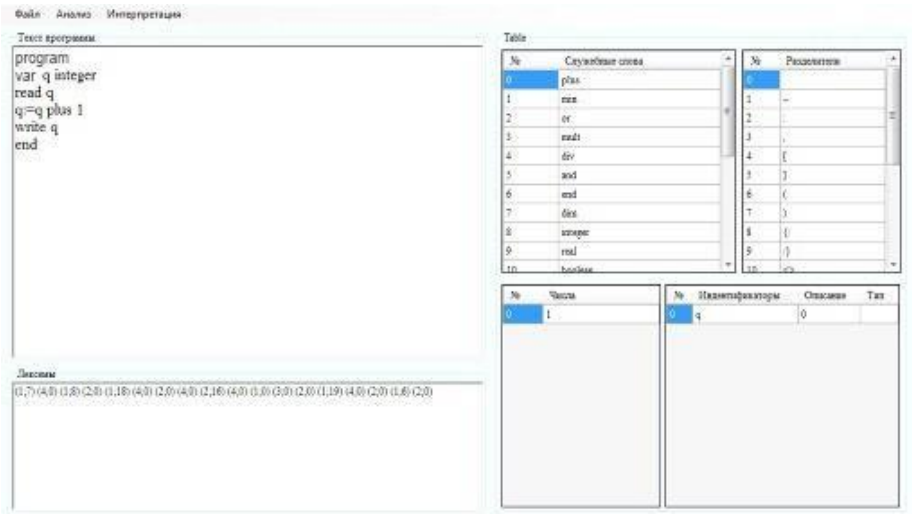
<операции_группы_отношения>:: = != | = = | < | <= | > | >=
<операции_группы_сложения>:: = + | - | ||
<операции_группы_умножения>:: = * | / | &&
<унарная_операция>::= !
<программа>::= program var <описание> begin <оператор> {; <оператор>} end.
<описание>::= <тип> <идентификатор> { , <идентификатор> }
<тип>::= int | float | bool
<оператор>::= <составной> | <присваивания> | <условный> | <фиксированного_цикла> | <условного_цикла> | <ввода> | <вывода>
<составной>::= begin <оператор> { ; <оператор> } end
<присваивания>::= <идентификатор> := <выражение>
<условный>::= if «(»<выражение> «)» <оператор> [else <оператор>]
<фиксированного_цикла>::= for <присваивания> to <выражение> [step <выражение>] <оператор> next
<условного_цикла>::= while «(»<выражение> «)» <оператор>
<ввода>::= readln идентификатор {, <идентификатор> }
<вывода>::= writeln <выражение> {, <выражение> }
<признак начала комментария>::= {
<признак конца комментария>::= }
<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*
<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
<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*)
<действительное>::= <числовая_строка><порядок> | [<числовая_строка>] . <числовая_строка> [порядок]
<числовая_строка>::= {/ <цифра> /}
<порядок>::= (E | e)[+ | -] <числовая_строка>

Формальные грамматики

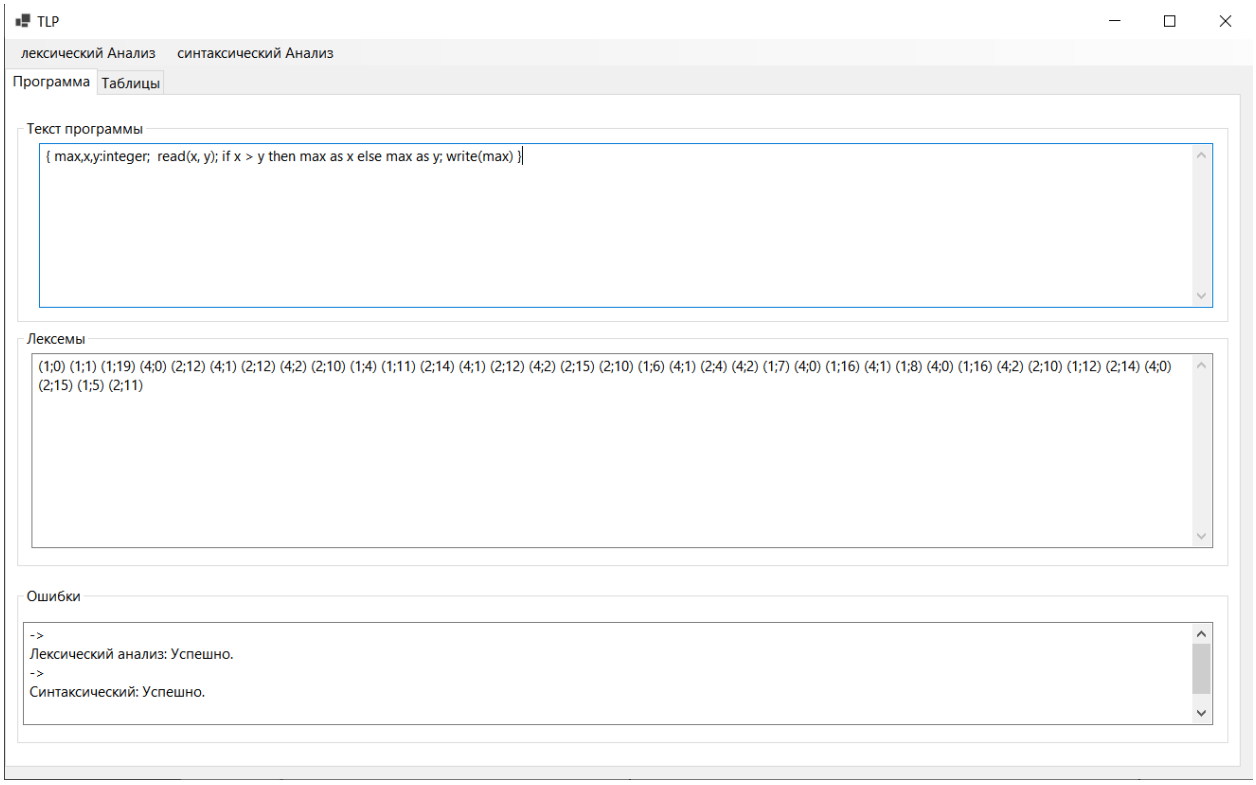
Pr→program var Opis begin Oper {; Oper}end.
Opis→Tip Ident{, Idet}
Oper→sostav | prisv | usl | fix | uslc | vvod|vyvod
Tip → int | float | bool
sostav→ begin Oper {; Oper} end
prisv→ Ident := vyraz
usl → if (Vyraz) Oper [else Oper]
fix → for prisv to vyraz [step vyraz] Oper next
uslc→ while (vyraz) Oper
vvod→ readln Ident {, Ident}
vyvod → writeln vyraz {, vyraz}
start_kom →{
end_kom →}
vyraz →Operand {otnosh Operand}
Operand→ slag {gr_slog slag}
slag→ mnog{umnog mnog}
mnog → Ident | chislo| const |unar mnog| (vyraz)
chislo → celoe | deystv
const → true | false
Ident → bookva{bookva| cifra}
otnosh → =!=| == | < | <= | > | >=
slog →+ | - | ||
umnog → * | / | &&
unar → =!
bookva → *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*
cifra→ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
celoe → dvoichnoe | vosmerich | desyatich | shestnad
dvoichnoe → dvc B| dvc b
dvc → 0 | 1 | dvc 0 | dvc 1
vosmerich → vosmc O | vosmc o
vosmc → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | vosmc0 | vosmc1 | vosmc2 | vosmc3 | vosmc4 | vosmc5 | vosmc6 | vosmc7
desyatich → cifraD | cifrad| cifra
shestnad1 → cifra | shestnsd1 cifra | shestnad1 buk16
shestnad → shestnad1H | shestnad1h
shestnad1Buk16→ A | B | C | D | E | F | a | b | c | d | e | f
deystvit → chisl_str Porydok | . Chisl_str | Chisl_ str. Chisl_str | Chisl_ str. Chisl_str Porydok | .chisl_str Porydok

Chisl_str→ Cifra | Chisl_str Cifra
Porydok→ E Znak Chisl_str | e Znak Chisl_str | E Chisl_str | e Chisl_str
Znak→ + | -

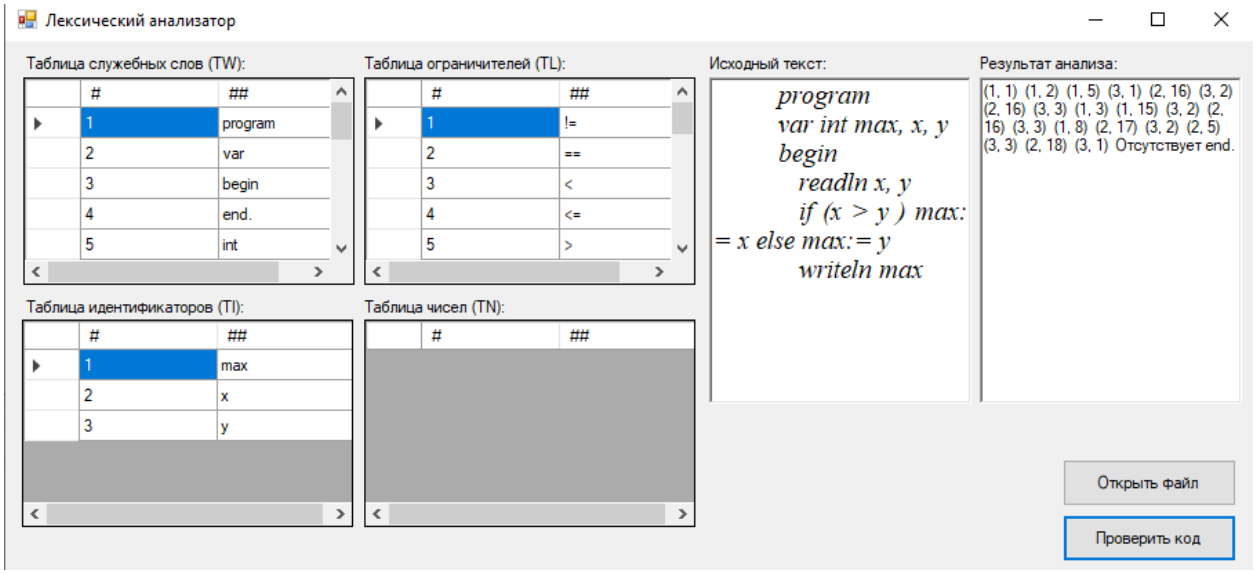
Вариант программы



Вариант программы 2



Вариант программы 3



Вариант программы 4