Грамматика

Белецкий Иван, Кабиров Роман

7 января 2023

1 Программа

$\langle program \rangle ::= \langle struct \rangle\ \langle program \rangle \mid \langle func \rangle\ \langle program \rangle \mid \langle var\ definition \rangle\ \langle program \rangle \mid type\ main(\langle parameters \rangle)\langle block \rangle$

2 Выражение

eps - nothing

$\langle expr \rangle ::= \langle expr \rangle \langle bin\ op \rangle \langle expr \rangle \mid \langle left\ un\ op \rangle \langle expr \rangle \mid \langle expr \rangle \langle right\ un\ op \rangle \mid \langle id \rangle \mid \langle func\ call \rangle \mid \langle lit \rangle \mid \langle id \rangle [\langle expr \rangle] \mid$
$\langle expr \rangle\ ?\ \langle expr \rangle : \langle expr \rangle \mid (\langle expr \rangle) \mid \langle id \rangle . \langle id \rangle$

$\langle bin\ op \rangle ::= \langle math\ bin\ op\ 1 \rangle \mid \langle math\ bin\ op\ 2 \rangle \mid \langle math\ bin\ op\ 3 \rangle \mid \langle equal\ bin\ op \rangle \mid$

$\langle logical\ bin\ op\ 1 \rangle \mid \langle logical\ bin\ op\ 2 \rangle \mid$

$\langle math\ bin\ op\ 1 \rangle ::= * * \mid \%$

$\langle math\ bin\ op; 2 \rangle ::= / \mid *$

$\langle math\ bin\ op; 3 \rangle ::= + \mid - \mid >= \mid <= \mid == \mid ! =$

$\langle equal\ bin\ op \rangle ::== \mid + = \mid - = \mid * = \mid / = \mid \% =$

$\langle logical\ bin\ op\ 1 \rangle ::= \&\&$

$\langle logical\ bin\ op\ 2 \rangle ::= || \mid \wedge$

$\langle left\ un\ op \rangle ::= ! \mid + + \mid - - \mid - +$

$\langle right\ un\ op \rangle ::= + + \mid - - \mid !$

$\langle id \rangle ::= \langle letter \rangle \mid \langle id \rangle \langle letter \rangle \mid \langle id \rangle \langle digit \rangle$

$\langle letter \rangle ::= a \mid b \mid c \mid \ldots \mid Z \mid \_$

$\langle digit \rangle ::= 0 \mid \ldots \mid 9$

$\langle func\ call \rangle ::= \langle id \rangle\ (\langle expressions \rangle);$

$\langle expressions \rangle ::= \langle expr \rangle,\ \langle expressions \rangle \mid \langle expr \rangle$

$\langle lit \rangle ::= true \mid false \mid \langle int\ lit \rangle \mid \langle real\ lit \rangle \mid \langle string \rangle$

$\langle int\ lit \rangle ::= \langle digit \rangle \langle int\ lit \rangle \mid \langle digit \rangle$

⟨*real lit*⟩ ::= ⟨*int lit*⟩.⟨*int lit*⟩

⟨*string*⟩ ::= "⟨*string lit*⟩"

⟨*character*⟩ ::= *any symbol*

⟨*char*⟩ ::=′ ⟨*char*⟩′

⟨*string lit*⟩ ::= ⟨*string lit*⟩⟨*character*⟩

3 Операторы

⟨*operator*⟩ ::= ⟨*expr*⟩; | ⟨*if*⟩ | ⟨*in*⟩ | ⟨*out*⟩ | ⟨*while*⟩ | ⟨*for*⟩ | ⟨*block*⟩ | *return* ⟨*expr*⟩; | *return*; | *break*; | *continue*; | ⟨*var definition*⟩
⟨*if*⟩ ::= *if* (⟨*expr*⟩) ⟨*operator*⟩

⟨*while*⟩ ::= *while* (⟨*expr*⟩) ⟨*operator*⟩

⟨*do while*⟩ ::= *do* ⟨*block*⟩ *while* (⟨*expr*⟩);

⟨*for*⟩ ::= *for* (⟨*expr*⟩; ⟨*expr*⟩; ⟨*expr*⟩) ⟨*operator*⟩

⟨*block*⟩ ::= {⟨*operators*⟩} | {}

⟨*var definition*⟩ ::= ⟨*type*⟩ ⟨*object names*⟩;

⟨*object names*⟩ ::= ⟨*id*⟩, ⟨*obect names*⟩ | ⟨*id*⟩ | *id* = ⟨*expr*⟩ | *id* = ⟨*expr*⟩, ⟨*object names*⟩ | *id*[⟨*int lit*⟩] | *id*[⟨*int lit*⟩], ⟨*object names*⟩

⟨*operators*⟩ ::= ⟨*operator*⟩; | ⟨*operator*⟩; ⟨*operators*⟩

⟨*in*⟩ ::= *in* (⟨*expressions*⟩);

⟨*out*⟩ ::= *out* (⟨*expressions*⟩);

4 Функция

⟨*func definition*⟩ ::= ⟨*type*⟩ ⟨*id*⟩ (⟨*parameters*⟩); | ⟨*type*⟩ ⟨*id*⟩ ();

⟨*func*⟩ ::= ⟨*type*⟩ ⟨*id*⟩ (⟨*parameters*⟩) ⟨*block*⟩ | ⟨*type*⟩ ⟨*id*⟩ ()⟨*block*⟩

⟨*type*⟩ ::= *int* | *float* | *bool* | *void* | ⟨*id*⟩

⟨*parameters*⟩ ::= ⟨*parameter*⟩ | ⟨*parameter*⟩, ⟨*parameters*⟩

⟨*parameter*⟩ ::= ⟨*type*⟩⟨*id*⟩

5 Структура

⟨*struct*⟩ ::= *struct* ⟨*id*⟩ {⟨*definitions*⟩} ⟨*object names*⟩; | *struct* ⟨*id*⟩ {⟨*definitions*⟩};

⟨*definitions*⟩ ::= ⟨*var definition*⟩; ⟨*definitions*⟩ | ⟨*var definition*⟩; | ⟨*func*⟩ ⟨*definitions*⟩ | ⟨*func definition*⟩ ⟨*definitions*⟩ | *eps*