Lab1b – Geanovu Medeea-Elena 933/1

Lexic.txt

Specifications

Alphabet:

a. Upper (A-Z) and lower case letters (a-z) of the English alphabet

b. Underline character '_';

c. Decimal digits (0-9);

Lexic:

a.Special symbols

- operators:

Arithmetic: + - * / % ++ --

Assigment: = += -=

Comparison: == != > < >= <=

Logical: && || ! & | ^

Sequence : ,

- separators () {} [] ; :

- reserved words:

abstract->is_purple_real

assert->purple_vibe

block->yellow

boolean->purplestance

break->purple_overload

case->shade

catch->purple_eject

char->lilac

const->purple_life

continue->purple_on_soldier

default->last_season

do->purple_order

double->orchid

else->purple_not

float->lavander

for->purple_roll

if->purple_check

int->violet

long->extra_purple

printLine->purple_out

readLine->purple_in

return->desaturated

static->dusk_purple

switch->purple_arsenal

this->their_purple_highness

throw->purple_enemy

try->purple_friend_check

tuple->tone_friends

void->outerspace

while->perpetual_purpling

b.identifiers

-a sequence of letters and digits, the first character may be "_", the rule is:

identifier = ["_"] letter | digit {letter|digit}

letter = "A" | "B" | ... | "Z" | "a" | "b"| … | "z"

digit = "0" | "1" | … | "9"

non_zero_digit = "1" |...| "9"

zero_digit = "0"

sign = "+" | "-"

whitespace = " " | "\n"

emptystring = ""

c.constants

1. integer - rule:

Integer = zero_digit | [sign] non_zero_digit {digit}

2.character

character = ‘letter’ | ‘digit’

3. string

string = "string"

string = character | {string | whitespace| emptystring}

4. defined constants

freezing_nightsky = -2,147,483,648 (*minimum int value*)

2. Syntax:

Syntax.in

The words - predefined tokens are specified between " and ":

Syntactical rules:

program = statement_list

compound_statement = ”{” statement_list ”}”

statement_list = statement ”;” {statement”}

statement = declaration_statement | assign_statement | struct_statement

declaration_statement = type identifier

type = usual_type | array_type

usual_type = ”violet” | ”lilac” | ”lavander” | ”orchid”

array_type = usual_type ”[” number of element ”]”

simple_statement = assign_statement | io_statement

assign_statement = IDENTIFIER ”=” expression

expression = [!] term | expression operation expression

term = factor | term operation factor

factor = IDENTIFIER | CONSTANT | "(" expression ")"

operation = “+” | “-” | “*” | ”/” | “%” | “^” | “&” | “|” (*The above mentioned operators*)

io_statement = "purple_in" | "purple_out" "(" IDENTIFIER | CONSTANT ")";

struct_statement = compound_statement | while_statement | if_statement | for_statement | switch_statement

while_statement = ”perpetual_purpleing” ”(” condition ”)” ”{” statement | compound_statement ”}”

if_statement = "purple_check" ”(” condition ”)” ”{” statement | compound_statement ”}” "purple_not" ”{” statement | compound_statement}”}”

for_statement = ”purple_roll” ”(” assign_statment ”;” condition ”;” {assign_statement} “)” ”{” statement | compound_statement ”}”

switch_statement = ”purple_arsenal” ”(” condition ”)” case_statement {case_statement} “last_season” ”:” statement_list

case_statement = ”shade” ”:” statement_list ”purple_overload”;

condition = expression RELATION expression

RELATION = "<" | "<=" | "=" | “!=” | "<>" | ">=" | ">" | “&&” | “||”

Tokens.in

[

]

{

}

;

:

,

<

<=

>=

==

!=

!

&&

&

^

=

+

-

*****

%
/

is_purple_real

purple_vibe

yellow

purplestance

shade

purple_eject

lilac

purple_life

purple_on_soldier

last_season

purple_order

orchid

purple_not

lavander

purple_roll

purple_check

violet

extra_purple

purple_out

purple_in

desaturated

dusk_purple

purple_arsenal

their_purple_highness

purple_enemy

purple_friend_check

tone_friends

outerspace

perpetual_purpling