**MLP din C++**

1. Specificare MLP

Elemente lexicale:

Operatori: +, -, \* , %, /, =, >, <, ==, !=

Delimitatori/separatori: “;”, “{”, “}”, “[”, “]”, “(”, “)”, “,”, “.”, “<<”, “>>”, “ ”

Cuvinte rezervate/cheie: #include , <iostream>, using, namespace, std, int, main(), double, cin, cout, return, if, while

Identificatori:

ID ::= caracter | ID {caracter} {cifra}

CONST ::= int | double

MESAJ ::= ‘ “ ’ {caracter\_mesaj} ‘ ” ’

int ::= 0 | ([‘+’|‘-’]cifra\_nenula{cifra})

double ::= int [‘.’{cifra}]

cifra ::= 0 | 1 | 2 | … | 9

cifra\_nenula ::= 1 | 2 | … | 9

caracter ::= ‘a’ | ‘b’ | ‘c’ | … | ‘z’ | ‘A’ | ‘B’ | ‘C’ | … | ‘Z’ | ‘\_’

caracter\_mesaj ::= ‘ ’ | ‘=’ | ‘+’ | ‘.’ | ‘,’ | ‘:’ | caracter

semn ::= ‘+’ | ‘-‘ | ‘\*’ | ‘/’ | ‘%’

comparator ::= ‘<’ | ‘>’ | “==” | “!=”

Limbaj(specificare BNF):

<program> ::= “#include” “<iostream>” “using” “namespace” “std” ‘;’ “int” “main()” ‘{‘ <lista\_instructiuni> <return> ‘}’

<lista\_instructiuni> ::= <instructiuni> | <instructiuni> <lista\_instructiuni>

<instructiuni> ::= <lista\_declaratii> | <lista\_operatii>

| <lista\_declaratii> <lista\_operatii>

<lista\_declaratii> ::= <declaratie> | <declaratie> <lista\_declaratii>

<lista\_operatii> ::= <operatie> | <operatie> <lista\_operatii>

<declaratie> ::= <tip> <nume\_variabile> ‘;’

<tip> ::= “int” | “double”

<nume\_variabile> ::= <nume> | <nume> ‘,’ <nume\_variabile>

<nume> ::= ID | ID ‘[’ int ‘]’

<operatie> ::= <atribuire> | <intrare> | <iesire> | <instr\_rel> | <instr\_ciclare>

<atribuire> ::= ID ‘=’ <expr> ‘;’

<expr> ::= <expr> semn <expr> | CONST | ID

<intrare> ::= “cin” “>>” ID ‘;’

<iesire> ::= “cout” “<<” <date\_iesire> ‘;’

<date\_iesire> ::= <expr> | MESAJ

<instr\_rel> ::= “if” <relatie> ‘{’ <lista\_instructiuni> ‘}’

<instr\_ciclare> ::= “while” <relatie> ‘{’ <lista\_instructiuni> ‘}’

<relatie> ::= ‘(’ <elem\_relatie> comparator <elem\_relatie> ‘)’

<elem\_realte> ::= <relatie> | CONST | ID

<return> ::= “return” const ‘;’

1. Texte sursa mini-programe care respecta specificatiile MLP:

|  |  |  |
| --- | --- | --- |
| P1 - calculeaza perimetrul si aria cercului de o raza data data | P2 - determina cmmdc a 2 nr. naturale | P3 - calculeaza suma a n numere citite de la tastatura |
| #include <iostream>  using namespace std ;  int main() {  double r ;  double PI ;  PI = 3.14 ;  cout << "Raza=" ;  cin >> r ;    cout << "Perimetrul=" ;  cout << 2 \* PI \* r ;  cout << "Aria=" ;  cout << PI \* r \* r ;  return 0 ;  } | #include <iostream>  using namespace std ;  int main() {  int a ;  int b ;  int rest ;  cout << "a=" ;  cin >> a ;  cout << "b=" ;  cin >> b ;  while ( b > 0 ) {  r = a % b ;  a = b ;  b = rest ;  }  cout << "CMMDC=" ;  cout << a ;  return 0 ;  } | #include <iostream>  using namespace std ;  int main() {  int suma ;  int n ;  int nr ;  int i ;    i = 0 ;  suma = 0 ;  cout << "n=" ;  cin >> n ;  while ( i < n ) {  cout << "nr=" ;  cin >> nr;  suma = suma + nr ;  i = i + 1 ;  }  cout << "Suma=" ;  cout << suma ;  return 0 ;  } |

1. Texte sursa mini-programe cu erori

|  |  |
| --- | --- |
| - Unul dintre programe contine doua erori care sunt in acelasi timp erori in limbajul original | - Al doilea program contine doua erori conform MLP, dar care nu sunt erori in limbajul original. Se cere ca acesta sa fie compilat si executat in limbajul original ales. |
| #include <iostream>  using namespace std ;  int main() {  int a ;  int b ;  int c[2] ;  a = 4 ;  b = 6 *-error: sintaxa gresita, lipseste ‘;’*  a + b = 10 ; *-error: in stanga trebuie sa fie o singura variabile care sa genereze referinta la un obiect*    return 0 ;  } | #include <iostream>  using namespace std ;  int main() {  int a ;  int b ;    cin >> a >> b ;  *-error: in secificarea MLP nu este permisa citirea multipla, intrarea mai multor date de la un cin*  if ( a > b ) {  a = 0 ;  }  else {  a = b – a ;  } *-error: in specificarea MLP nu exista optiunea else pentru if*  a++ ; *-error: nu exista operatorul “++” in specificarea MLP*    cout << a << endl ;  *-error: nu exista “endl” in secificarea MLP*  *-error: in secificarea MLP nu este permisa afisarea multipla, iesirea mai multor date de la un cout*  return 0 ;  } |