1. Specificarea minilimbajului (BNF):

|  |  |
| --- | --- |
| MyMLP | MLP subset of C++ |
| <program> → birth{ <lista\_instr> die.}  <lista\_instr> → <instr> | <instr> ; <lista\_instr>  <instr> → <atribuire>  <instr> → <instr\_intrare> | <instr\_iesire>  <instr> → <instr\_if> | <instr\_for> | <instr\_while>  <declarare> → ID : <tip>  <tip> → integer | double | circle  <atribuire> → ID <operator\_atribuire> <operand>  <operator\_atribuire> → =  <operand> → CONST | ID | <expr>  <expr> → <operand> <operator\_aritmetic> <operand>  <operator\_aritmetic> → + | - | \* | ++  <operator\_unar> → ++ | --  <expr> → <operand>  <instr\_if> → supposing (<conditie>) thereupon <atribuire> | supposing (<conditie>) {<lista\_instructiuni>}  <instr\_intrare> → gather -> ID  <instr\_iesire> → dash <- ID  <instr\_ciclare> → purge (<atribuire>; <conditie >; <pas>) {<lista\_instr>}  <pas> → ID<operator\_unar>  <instr\_while> → given (<conditie>) {<lista\_instr>}  <conditie > → <operand> <operator\_relational> <operand>  <operator\_relational> → < | > | != | <= | <program> → int main(){ <lista\_instr> }  <lista\_instr> → <instr> | <instr> ; <lista\_instr>  <instr> → <atribuire>  <instr> → <instr\_intrare> | <instr\_iesire>  <instr> → <instr\_if> | <instr\_for> | <instr\_while>  <declarare> → ID <tip>  <tip> → int | double | circle  <atribuire> → ID <operator\_atribuire> <operand>  <operator\_atribuire> → =  <operand> → CONST | ID | <expr>  <expr> → <operand> <operator\_aritmetic> <operand>  <operator\_aritmetic> → + | - | \* | ++  <operator\_unar> → ++ | --  <expr> → <operand>  <instr\_if> → if (<conditie>) {<lista\_instructiuni>}  <instr\_intrare> → cin >> ID  <instr\_iesire> → cout << ID  <instr\_ciclare> → for (<atribuire>; <conditie >; <pas>) {<lista\_instr>}  <pas> → ID<operator\_unar>  <instr\_while> → while (<conditie>) {<lista\_instr>}  <conditie > → <operand> <operator\_relational> <operand>  <operator\_relational> → < | > | != | <= |

1. Mini programe

|  |  |  |
| --- | --- | --- |
| C++ | MLP subset of C++ | MyMLP |
| Perimetrul si aria cercului | | |
| #include <iostream>  /\*  computes and displays the perimeter and area of a circle  in: radius – double ; the radius of the circle  out: -  console outputs the perimeter and area  \*/  int main() {  double radius;  std::cin >> radius;  double perimeter;  double area;  double PI = 3.14;  perimeter = 2 \* PI \* radius;  area = PI \* radius \* radius;  std::cout << perimeter << "\n";  std::cout << area;  return 0;  } | int main() {  double radius;  cin >> radius;  double perimeter;  double area;  double PI;  PI = 3.14;  perimeter = 2 \* PI \* radius;  area = PI \* radius \* radius;  cout << perimeter;  cout << area;  return 0;  } | birth{  radius: double;  gather => radius;  perimeter: double;  area: double;  PI: double;  PI = 3.14;  perimeter = 2 \* PI \* radius;  area = PI \* radius \* radius;  dash <- perimeter;  dash <- area;  die.  } |
| Cmmdc 2 numere | | |
| #include <iostream>  /\*  computes and displays the greatest common divisor of two numbers  in: a,b - integer ; the two numbers  out: -  console outputs the gcd of the given numbers  \*/  int main() {  int a;  int b;  std::cin >> a;  std::cin >> b;  while (a != b)  {  if (a > b)  a = a - b;  else  b = b - a;  }  std::cout << a;  return 0;  } | int main() {  int a;  int b;  cin >> a;  cin >> b;  while (a != b)  {  if (a > b)  a = a - b;  if (a <= b)  b = b - a;  }  cout << a;  return 0;  } | birth{  a: integer;  b: integer;  gather -> a;  gather -> b;  given(a != b) {  supposing(a > b) thereupon a = a - b;  supposing(a <= b) thereupon b = b - a;  }  dash <- a;  die.  } |
| Suma a n numere | | |
| #include <iostream>  /\*  computes and displays the sum of n given numbers  in: x1,x2..,xn - integer  out: -  console outputs the sum of the n numbers  \*/  int main() {  int s = 0;  int n;  int x;  std::cin >> n;  for (int i = 1; i <= n; i++) {  std::cin >> x;  s = s + x;  }  std::cout << s;  return 0;  } | int main() {  int s;  s = 0;  int n;  int x;  cin >> n;  for (int i = 1; i <= n; i++) {  cin >> x;  s = s + x;  }  cout << s;  return 0;  } | birth{  s: integer;  s = 0;  n: integer;  x: integer;  i: integer;  gather => n;  purge(i = 0; i < n; i++) {  gather -> x;  s = s + x;  }  dash <- s;  die.  } |

1. Erori

* Erori din MLP care sunt erori si in C++ (lipsa separator dupa instructiune, lipsa acolada de inceput de bloc de instructiuni pentru for)

|  |  |
| --- | --- |
| MLP subset of C++ | MyMLP |
| int main() {  int s  s = 0;  int n;  int x;  cin >> n;  for (int i = 1; i <= n; i++)  cin >> x;  s = s + x;  }  cout << s;  return 0;  } | s: integer  s = 0;  n: integer;  x: integer;  i: integer;  gather -> n;  purge(i = 0; i < n; i++)  gather -> x;  s = s + x;  }  dash <- s; |

* Erori din MLP care nu sunt erori in C++ (nume de variabila > 8 caractere, instructiunea else nu e definita in MLP)

|  |  |
| --- | --- |
| MLP subset of C++ | MyMLP |
| int main() {  int a12345678;  int b;  cin >> a12345678;  cin >> b;  while (a12345678 != b)  {  if (a12345678 > b)  a12345678 = a12345678 - b;  else  b = b - a12345678;  }  cout << a12345678;  return 0;  } | a12345678: integer;  b: integer;  gather -> a12345678;  gather -> b;  given(a != b) {  supposing(a12345678 > b) thereupon a12345678 = a12345678 - b;  else b = b - a12345678;  }  dash <- a12345678; |