**Tema 1:** [**https://www.cs.ubbcluj.ro/~dana/LFTC/Resurse/tema1\_lab.pdf**](https://www.cs.ubbcluj.ro/~dana/LFTC/Resurse/tema1_lab.pdf)

|  |  |  |  |
| --- | --- | --- | --- |
| **3** | **aac** | **Patras Sergiu Adrian** | **patras.sergiu@yahoo.com** |

Ex2:

- calculeaza perimetrul si aria cercului de o raza data data

public double perimetruCerc(int r) {  
 double rez = 2 \* Math.*PI* \* r;  
 return rez;  
}  
  
public double ariaCerc(int r) {  
 double rez = Math.*PI* \* r \* r;  
 return rez;  
}

- determina cmmdc a 2 nr naturale

public int cmmdc(int a, int b) {  
 if(a \* b == 0)  
 return a+b;  
  
 while(a != b) {  
 if(a > b) {  
 a = a - b;  
 }  
 if(a < b) {  
 b = b - a;  
 }  
 }  
 return a;  
}

- calculeaza suma a n numere citite de la tastatura

public int suma(int n) {  
 int sum = 0, a;  
 while(n > 0) {  
 a = scanner.nextInt();  
 sum = sum + a;  
 n = n - 1;  
 }  
 return sum;  
}

Ex1:

- 2 tipuri de date simple si un tip de date definit de utilizator

<tipuri\_simple> ::= int, double

<tip\_user> ::= class Tema

- instructiuni:

- o instructiune de atribuire *✓*

- o instructiune de intrare/iesire *✓*

- o instructiune de selectie (conditionala) *✓*

- o instructiune de ciclare *✓*

**Specificarea minilimbajului de programare (MLP)**

<declarative> ::= <type> ID = <termen>;

<type> ::= int | double | class

<clasa> ::= public class ID {<declarative> <constructor> <functii> }

<constructor> ::= ID { <lista\_instructiuni> }

<functii> ::= <functie> | <functie> <functii>

<functie> ::= <antet> { <lista\_inst> }

<lista\_instr> ::= <instr> | <instr> ; <lista\_instr> ;

<opAritmetici> ::= “+” | “-“ | “\*”

<opLogici> ::= “<” | “>” | “!=”

<operatori> ::= <opLogici> | <opAritmetici>

<instr> ::= <atribuire>  | <instr\_read> | <instr\_if> | <instr\_while> | <instr\_return>

<atribuire> ::= ID = <expr> | <instr\_read> | <instantiere>

<instantiere> ::= new CLASS\_ID(<parametrii\_clasa>)

<parametrii\_clasa> ::= < termen>

<expr> ::= <expr> <operatori> <termen> | <termen>

<termen> ::= ID | CONST

<instr\_read> ::= scanner.nextInt();

<instr\_if> ::= if ( <expr> ) { <lista\_instr> }

<instr\_while> ::= while ( <expr> ) { <lista\_instri> }

<instr\_return> ::= return ID | CONST;

ID: \b[a-z]\w{0,7}\b

Ex3:

**- Unul dintre programe contine doua erori care sunt in acelasi timp erori in limbajul original (pentru care MLP defineste un subset)**

public int suma(int n) {  
 int sum = 0  
 while n > 0 {  
 sum = sum + scanner.nextInt();  
 n = n - 1;  
 }  
 return sum;  
}

Erori:

- fara ; la finalul unei instructiuni

- la instructiunea de ciclare nu avem () in jurul expresiei

**- 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.**

public int cmmdc(int a, int b) {  
 if(a \* b == 0)  
 return a+b;  
  
 while(a != b) {  
 if(a > b) {  
 a -= b;  
 }  
 else {  
 b -= a;  
 }  
 }  
 return a;  
}

**Erori:**

* Instructiunea de selectie din MLP nu are definita o sintaxa care foloseste else
* MLP nu are definite expresia -=