Laborator 11

Prioritati pentru reguli. Functii

Cateva probleme rezolvate

1. Scrieti un program care sa calculeze lungimea unei liste, fara sa distruga lista.

Indicatie: se creeaza o copie a listei.

R:

```
(deffacts fapte (lista 1 2 3 4 5))

(defrule initiala
        (declare (salience 3))
        (lista $?a)
        (not (lista1 $?))
        (not (nr ?))
=>
        (assert (lista1 $?a))
        (assert (nr 0))
)

(defrule lungime
        (declare (salience 2))
        ?f1<-(lista1 ?x $?a)
        ?f2<-(nr ?y)
=>
        (retract ?f1)
        (retract ?f2)
        (assert (lista1 $?a))
        (assert (nr (+ ?y 1)))
)
```

```
(defrule finala
        (declare (salience 1))
        ?f<-(lista1 $?a)
        (nr ?)
=>
        (retract ?f)
)
```

2. Scrieti un program care genereaza toate permutarile unei multimi. Multimea se reprezinta ca o lista.

R:

```
(deffacts fapte (lista 1 2 3 4))

(defrule permutari
        (lista $?a ?x ?y $?b)
=>
        (assert (lista $?a ?y ?x $?b))
)
```

3. Scrieti un program care sa citeasca lungimea si latimea unui dreptunghi si care sa afiseze aria si perimetrul dreptunghiului. Pentru arie si perimetru se vor folosi 2 functii.

R:

```
(deffunction perimetru
        (?a ?b)
        (+ ?a ?a ?b ?b)
)

(deffunction arie
        (?a ?b)
        (* ?a ?b)
```

```
)

(defrule initiala
        (not (lungime ?))
        (not (latime ?))
=>
        (printout t "Lungimea = ")
        (assert (lungime (read)))
        (printout t "Latimea = ")
        (assert (latime (read)))
)

(defrule finala
        (lungime ?x)
        (latime ?y)
=>
        (printout t "Perimetrul este: " (perimetru ?x ?y) crlf "Aria este: " (arie ?x ?y) crlf)
)
```

4. Scrieti un program care sa citeasca un numar natural n si sa afiseze n!.

R:

```
(defrule initiala
        (declare (salience 3))
        (not (n ?))
        (not (fact ?))
        (not (cnt ?))
=>
        (printout t "n = ")
        (assert (n (read)))
        (assert (fact 1))
        (assert (cnt 1))
```

```
)

(defrule factorial
        (declare (salience 2))
        (n ?x)
        ?f<-(cnt ?y&:(<= ?y ?x))
        ?g<-(fact ?z)
=>
        (retract ?f)
        (retract ?g)
        (assert (cnt (+ ?y 1)))
        (assert (fact (* ?z ?y)))
)

(defrule finala
        (declare (salience 1))
        (n ?x)
        (fact ?y)
        ?f<-(cnt ?)
=>
        (retract ?f)
        (printout t ?x "!" = ?y crlf)
)
```

5. Scrieti un program care sa citeasca un numar natural n si sa afiseze al n-lea termen al sirului Fibonacci.

R:

```
(defrule initiala
        (declare (salience 3))
        (not (n ?))
        (not (fibo ? ?))
```

```
        (not (cnt ?))
=>
        (printout t "n = ")
        (assert (n (read)))
        (assert (fibo 0 1))
        (assert (cnt 1))
)

(defrule fibonacci
        (declare (salience 2))
        (n ?x)
        ?f<-(cnt ?y&:(<= ?y ?x))
        ?g<-(fibo ?z1 ?z2)
=>
        (retract ?f)
        (retract ?g)
        (assert (cnt (+ ?y 1)))
        (assert (fibo ?z2 (+ ?z1 ?z2)))
)

(defrule finala
        (declare (salience 1))
        (n ?x)
        (fibo ?y ?z)
        ?f<-(cnt ?)
=>
        (retract ?f)
        (printout t "fibo(" ?x ")" = ?y crlf)
)
```

6. Scrieti un program care sa citeasca 2 numere naturale m si n si sa afiseze cmmdc(m,n).

R:

```
(defrule initiala
        (declare (salience 3))
        (not (nr ? ?))
        (not (nr1 ? ?))
=>
        (printout t "Dati numerele: " crlf)
        (assert (nr (read) (read)))
        (assert (sablon))
)

(defrule initiala2
        (declare (salience 3))
        (nr ?x ?y)
        (not (nr1 ? ?))
        ?f<-(sablon)
=>
        (retract ?f)
        (assert (nr1 ?x ?y))
)

(defrule cmmdc
        (declare (salience 2))
        ?f<-(nr1 ?x ?y&:(> ?y 0))
=>
        (retract ?f)
        (assert (nr1 ?y (mod ?x ?y)))
)
```

```
(defrule finala
        (declare (salience 1))
        (nr ?x ?y)
        ?f<-(nr1 ?a ?b)
=>
        (retract ?f)
        (assert (cmmdc ?a))
        (printout t "cmmdc(" ?x "," ?y ")=" ?a crlf)
)
```