Seminar 6

PROBLEME CU ARBORI BINARI

1. Sa se construiasca arborele binar asociat unei expresii aritmetice continda operatorii +, -, /, *, pornind de la forma ei poatfixata

   Ex : (a+b) * c - (d + e * f) + g

   ab+c*def*+-g+

   ->folosim o stiva

   ->daca element curent este operand => adaugam in stiva nod cu informatie utila = element

   ->daca elem = operator

   ->creez un nod cu informatie utila = element

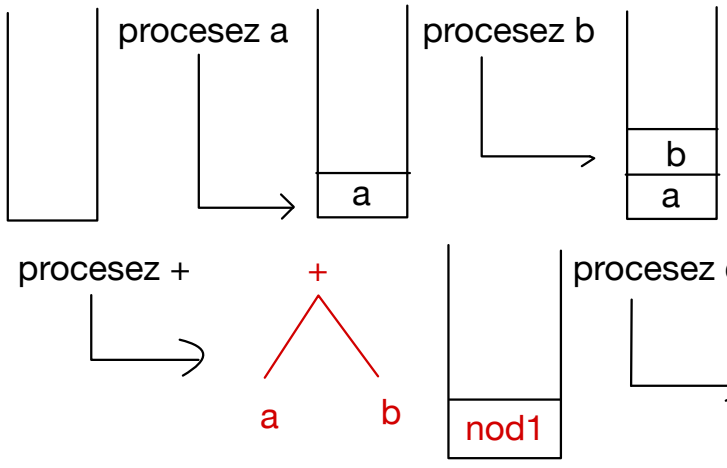
   ->sterg 2 elemente din stiva ("pop")

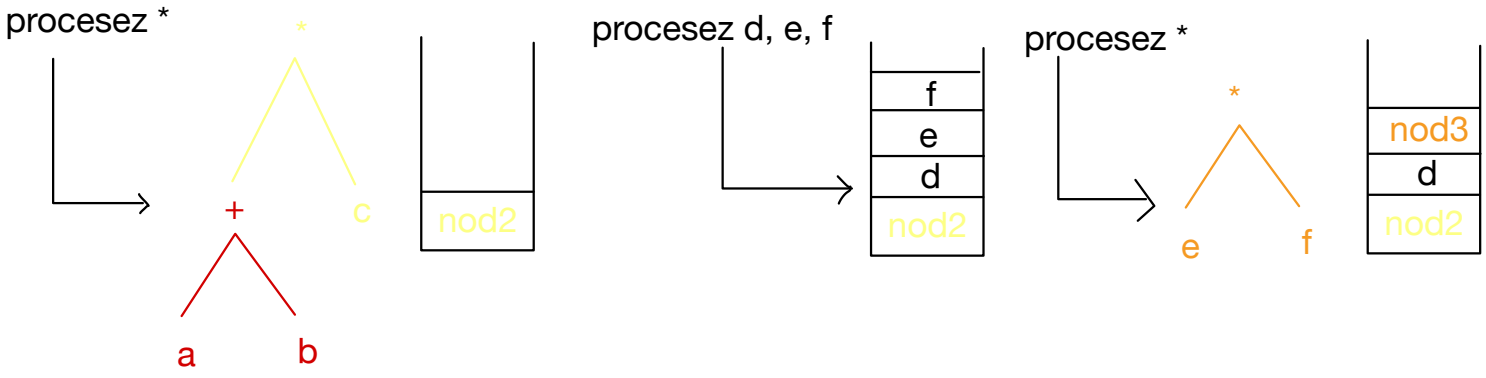
   ->primul element scos din stiva : fiu drep al nodului care contine operator

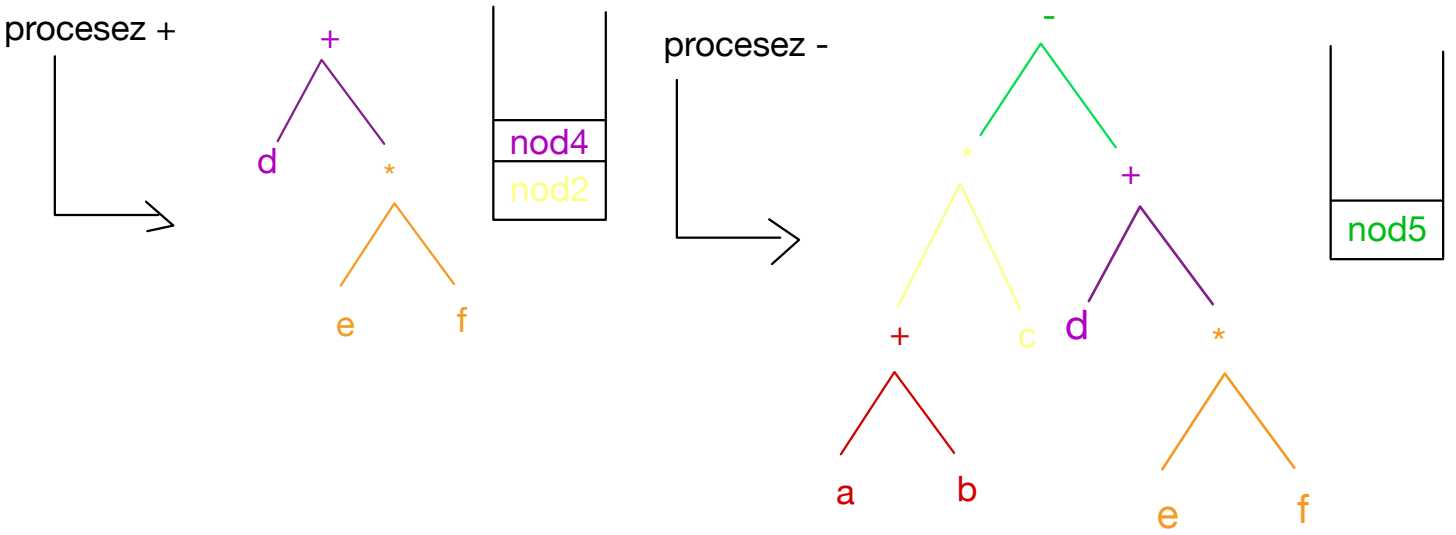
   ->al doilea element scos din stiva => fiu stang

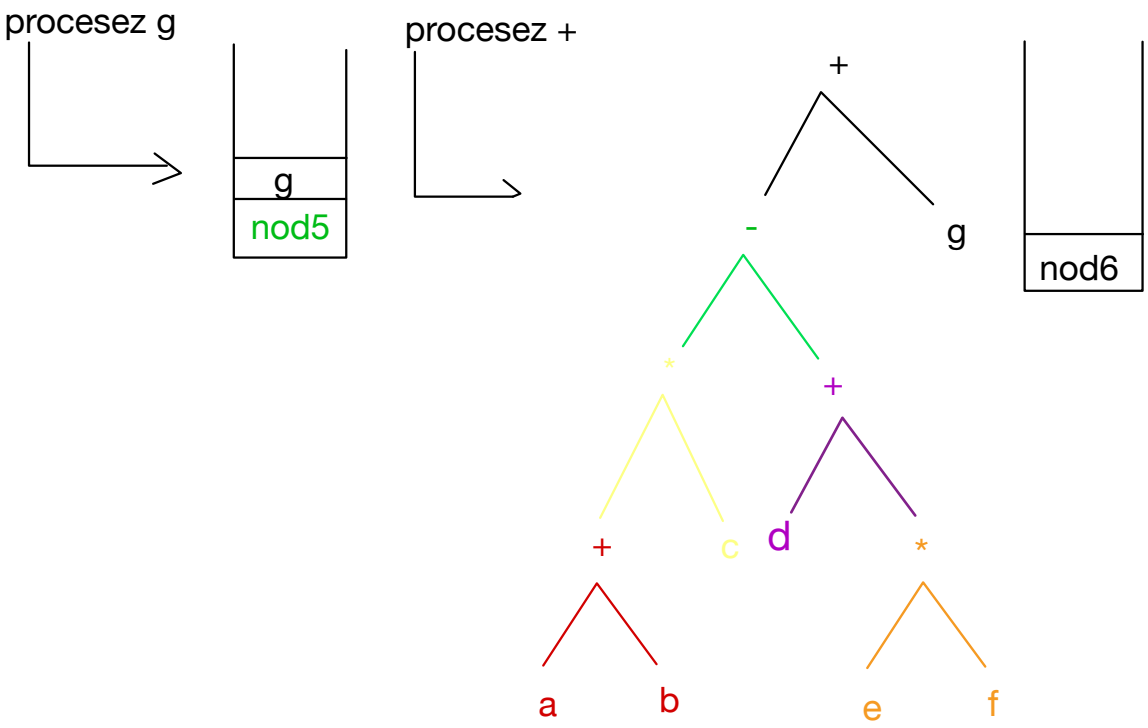
	Stiva	Coada
(	(	
a		a
+	(+	
b	(+	ab
)		ab+
*	*	
c	*	ab+c
-	-	ab+c*
(	-(	ab+c*
d	-(	ab+c * d
+	-(+	ab+c*de
e	-(+*	ab+c*de
*	-(+*	ab+c*de
f	-(+*	ab+c*def
)	-	ab+c*def*+
+	+	ab+c*def*+-
g	+	ab+c*def*+-g
		ab+c*def*+-g+

ab+e*def*+-g+









-> parcurgere in postordine -> forma postfixata

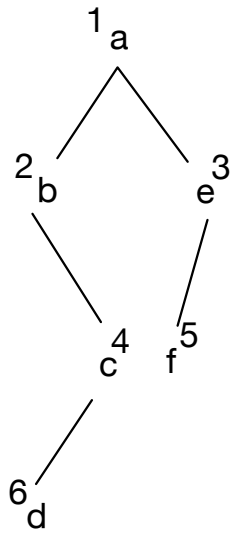
```
subalgoritm creeaza(Epost, arb)
    creeaza(s) //cream o stiva vida
    pentru fiecare e din Epost executa
        aloca(nodNou)
        [nodNou].e <- e
        daca e este operand atunci
            [nodNou].st <- NIL
            [nodNou].dr <- NIL
        altfel
            sterge(s,p1)
            sterge(s,p2)
            [nodNou].st <- p2
            [nodNou].dr <- p1
        sf daca
        adauga(s,nodNou)
    sf pentru
    sterge(s, p)
    arb.rad <- p
sf subalg
```

```
Nod
    e : TElement
    st, dr : *Nod

AB
    rad : *Nod
```

```
subalgoritm creeaza2(Epost, arb)
    creeaza(s)
    pentru fiecare e din Epost
        daca e este operand
            creeazaFrunza(ab, e)
        altfel
            sterge(s, p1)
            sterge(s, p2)
            creeazaArbore(ab, p2, e, p1)
        sf daca
        adauga(s, ab)
    sf. pentru
    sterge(s, arb)
```

2. Sa se genereze tabelul coresp. arborelui



	info utila	indice fiu stang	indice fiu drept
1	a	2	3
2	b	0	4
3	e	5	0
4	c	6	0
5	f	0	0
6	d	0	0

```
subalgoritm numerotare(arb, k)
    k <- 0
    creeaza(0)
    daca (arb.rad != NIL) atunci
        k <- 1
        [arb.rad].nr <- k
        adauga(c, arb.rad)
    sf daca
    cat timp (!vida(c)) executa
        sterge(c, p)
        daca [p].st != NIL atunci
            k <- k + 1
            [[p].st].nr <- k
            adauga(c, [p].st)
        sf daca
        daca [p].dr != NIL atunci
            k <- k + 1
            [[p].dr].nr <- k
            adauga(c, [p].dr)
        sf daca
    sf cat timp
sf subalg
```

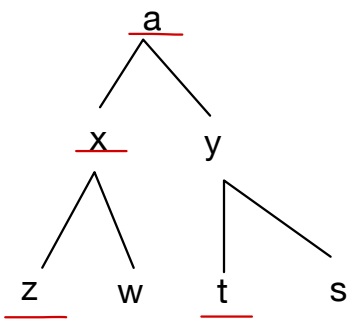
```
Nod
    e : TElement
    st, dr : *Nod
    nr : Intreg

AB
    rad : *Nod
```

```
subalgoritm parcurgere(p, T)
    daca p != NIL atunci
        T[[p].nr, 1] <- [p].e
        daca [p].st != NIL atunci
            T[[p].nr, 2] <- [[p].st].nr
        altfel
            T[[p].nr, 2] <- 0
        sf daca
        daca [p].dr != NIL atunci
            T[[p].nr, 3] <- [[p].dr].nr
        altfel
            T[[p].nr, 3] <- 0
        sf daca
        parcurgere([p].st, T)
        parcurgere([p].dr, T)
```

```
    sf daca
sf subalg
```

3. Se da arborele genealogic al unei persoane
arborescenta st -> linia materna
dr -> linia paterna
rad : gen feminin
a) Afisare toate pers de gen feminin



a, x, z, t

```
subalgoritm genFeminin(arb) :
    creeaza(c)
    daca arb.rad != NIL atunci
        scrie [arb.rad].e
        adauga(c, arb.rad)
    sf daca
    cat timp (!vida(c)) executa
        sterge(c, p)
        daca([p].st != NIL) atunci
            scrie[[p].st].e
            adauga(c, [p].st)
        sf daca
        daca ([p].dr != NIL) atunci
            adauga(c, [p].dr)
        sf daca
    sf cat timp
sf subalg
```