

# Tema 24

## Problemă

Se dă o listă de procese caracterizate prin nume, utilizare procesor (1-100%) și timp de rulare și numele unui proces din listă. Acestea vor fi preluate de către sistem și rulate în funcție de factorul de utilizare al procesorului (cele cu factor mai mic au prioritate). Să se afișeze după cât timp procesul cu numele dat va fi preluat de către sistem.

## TAD ListăOrdonată

$L = \{l \mid l = [e_1, e_2, \dots, e_n], e_i \in T \text{ Comparabil și } e_1 \leq e_2 \leq \dots \leq e_n, \forall i = 1, 2, \dots, n\}$

## Interfață

- creează ( $l$ )  
 pre : true  
 post :  $l \in L, l = \Phi$  lista vidă
- adaugă ( $l, e$ )  
 pre :  $l \in L, e \in T \text{ Comparabil}$   
 post :  $l' = (e_1, \dots, e_{i-1}, e, e_{i+1}, \dots, e_n) (e_{i-1} \leq e \leq e_{i+1})$
- șterge ( $l, i, e$ )  
 pre :  $l \in L, l = (e_1, \dots, e_{i-1}, e, e_{i+1}, \dots, e_n), i \in T \text{ Întreg}, i$  poziție validă  
 post :  $e \in T \text{ Comparabil}, e = \text{elementul de pe poziția } i \text{ din } l$   
 $l' = (e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_n)$   
 @ aruncă excepție dacă  $i$  nu e valid
- caută ( $l, e$ )  
 pre :  $l \in L, e \in T \text{ Comparabil}$   
 post :  $\text{caută} = \begin{cases} i, & \text{dacă } i \text{ e prima poziție pe care } e \text{ a fost găsit în lista } l \\ -1, & e \notin L \end{cases}$

- element (l, i, e)  
pre :  $l \in L, i \in T \text{ Întreg}, i$  poziție validă  
post :  $e \in T \text{ Comparabil}, e$  = elementul de pe poziția  $i$  din  $l$   
@ aruncă excepție dacă  $i$  nu e valid
- vidă (l)  
pre :  $l \in L$   
post :  $vidă = \begin{cases} true, & \text{dacă } l = \Phi \\ false, & \text{altfel} \end{cases}$
- dim (l)  
pre :  $l \in L$   
post :  $dim = n \in T \text{ Întreg},$   
 $n$  = numărul de elemente din lista  $l$
- iterator (l, i)  
pre :  $l \in L$   
post :  $i \in I, i$  este un iterator pe lista  $l$
- distruge (l)  
pre :  $l \in L$   
post :  $l$  a fost "distrușă" (spațiul de memorie alocat a fost eliberat)

## **TAD IteratorListăOrdonată**

$I = \{i \mid i \text{ este un iterator pe o listă ordonată având elemente de tip } T \text{ Comparabil}\}$

### **Interfață**

- creează (i, l)  
pre :  $l$  este o listă ordonată  
post :  $i \in I$ , s-a creat iteratorul  $i$  pe lista ordonată  $l$   
(elementul **curent** din iterator referă "primul" element din listă)
- element (i, e)  
pre :  $i \in I$ , **curent** este valid (referă un element din listă)  
post :  $e \in T \text{ Element}, e$  este elementul curent din iterație  
(elementul din listă referit de *curent*)

- valid (i)

pre :  $i \in I$

post :  $valid = \begin{cases} \text{adev\c{a}rat} , \text{dac\c{a} curent refer\c{a} o pozi\c{t}ie valid\c{a} din list\c{a} \\ \text{fals} , \text{altfel} \end{cases}$

- urm\c{a}tor (i)

pre :  $i \in I$ , *curent* este valid

post : *curent* refer\c{a} "urm\c{a}torul" element din list\c{a} fa\c{t}\c{a} de cel referit de *curent*

- distruge (i)

pre :  $i \in I$

post :  $i$  a fost "distru\c{s}" (spa\c{t}iul de memorie alocat a fost eliberat)

## Reprezentare

List\c{a}Ordonat\c{a}:

- list\c{a} dublu \c{a}n\c{l}\c{a}n\c{t}uit\c{a} cu \c{a}n\c{l}\c{a}n\c{t}uirile reprezentate pe tablou

IteratorList\c{a}Ordonat\c{a}:

- articol de forma {*curent*, *list\c{a}*}