

# Proiect

-  
pentru implementare

1. -ului.
  2. TAD independent de reprezentare. **Iterator** – pentru  
containerele care pot fi iterate.
  3. în pseudocod a
  4. la
  5. alegere,
  6. scrierea în pseudocod a
- D +

## Evaluarea proiectului

- 1p oficiu
- 1p
- 1.5p
- 2.75p
- 2.75p
- 1p stil

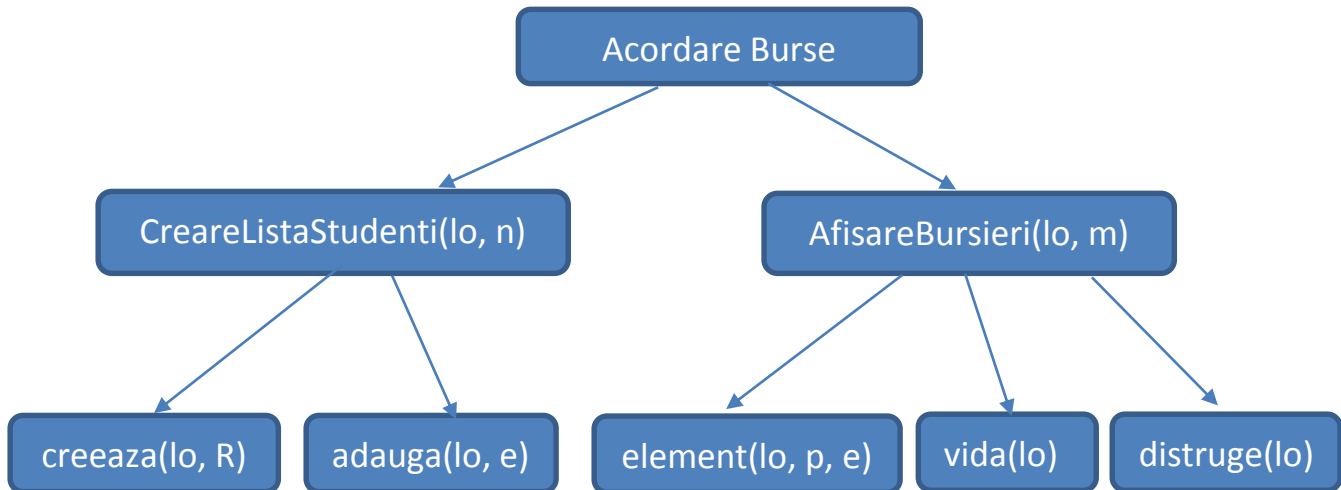
## TAD Lista Ordonată

### Problema

Se dau  $n$   
( $m \leq n$ )

$m$

## Diagrama de structura



# Sortări

## 1. Bucket Sort

-  $\epsilon[0, N - 1]$

-

Ideea:

-

- fiecare pereche (cheie,

complexitate =  $\theta(n)$

-

- complexitate =  $\theta(n + N)$

$$\Rightarrow \text{_____} \theta(n + N)$$

\_\_\_\_\_

- 
- )

## 2. Sortare lexicografică

### Ordine lexicografică

- Pentru d-tuplurile  $(x_1, x_2, \dots, x_d), (y_1, y_2, \dots, y_d)$ , avem:

$$(x_1, x_2, \dots, x_d) < (y_1, y_2, \dots, y_d) \Leftrightarrow (x_1 < y_1) \vee ((x_1 = y_1) \wedge (x_2, x_3, \dots, x_d) < (y_2, y_3, \dots, y_d))$$

- departe

- $C_i$  – tupluri folosind dimensiunea  $i$
- $stableSort(S, C)$   $C$ .

-  $stableSort$  de  $d$  ori.

Notând cu  $T(n)$  complexitatea algoritmului  $stableSort$  lexicografice devine  $\theta(d \cdot T(n))$ .

## 3. Radix Sort

- BucketSort  $\Rightarrow$
- complexitate:  $\theta(d \cdot (n + N))$