

TAD Stiva (STACK)

Observații:

1. În limbajul uzual cuvântul “stivă” referă o “grămadă în care elementele constitutive sunt așezate ordonat unele peste altele”.
 - Un element nou se adaugă în stivă deasupra elementului cel mai recent adăugat în stivă.
 - Din stivă se poate accesa și extrage doar elementul cel mai recent introdus.
 - Exemple de stive sunt multiple: stivă de farfurii, stivă de lemne, etc. Tipul de date **Stivă** permite implementarea în aplicații a acestor situații din lumea reală.
2. O *stivă* este o structură liniară de tip listă care restricționează adăugările și ștergerile la un singur capăt (listă LIFO - *Last In First Out*).
3. **Accesul** într-o stivă este *prespecificat* (se poate accesa doar elementul cel mai recent introdus în stivă - din vârful stivei), nu se permite accesul la elemente pe baza poziției. Dintr-o stivă se poate **șterge** elementul CEL MAI RECENT introdus în stivă - cel din vârful stivei.
4. Se poate considera și o capacitate inițială a stivei (număr maxim de elemente pe care le poate include), caz în care dacă numărul efectiv de elemente atinge capacitatea maximă, spunem că avem o *stivă plină*.
 - adăugarea în stiva plină se numește **depășire superioară**.
5. O stivă fără elemente o vom numi *stivă vidă* și o notăm Φ .
 - ștergerea din stiva vidă se numește **depășire inferioară**.
6. O stivă în general nu se iterează.
7. Stivele sunt frecvent utilizate în programare - recursivitate, backtracking iterativ.

Tipul Abstract de Date STIVA:

domeniu:

$$\mathcal{S} = \{s \mid s \text{ este o stivă cu elemente de tip } TElement\}$$

operații (interfața):

- **creeaza**(s)
{creează o stivă vidă}
 $pre : true$
 $post : s \in \mathcal{S}, s = \Phi(\text{stiva vidă})$

- $adauga(s, e)$
 {se adaugă un element în vârful stivei}
 $pre : s \in \mathcal{S}, e \in TElement, s \text{ nu e plină}$
 $post : s' \in \mathcal{S}, s' = s \oplus e, e \text{ va fi cel mai recent element introdus în stivă}$
 $\textcircled{!}$ aruncă excepție dacă stiva e plină
- $sterge(s, e)$
 {se scoate un element din vârful stivei}
 $pre : s \in \mathcal{S}, s \neq \Phi$
 $post : e \in TElement, e \text{ este cel mai recent element introdus în stivă}, s' \in \mathcal{S}, s' = s \ominus e$
 $\textcircled{!}$ aruncă excepție dacă stiva e vidă
- $element(s, e)$
 {se accesează elementul din vârful stivei}
 $pre : s \in \mathcal{S}, s \neq \Phi$
 $post : s' = s, e \in TElement, e \text{ este cel mai recent element introdus în stivă}$
 $\textcircled{!}$ aruncă excepție dacă stiva e vidă
- $vida(s)$
 $pre : s \in \mathcal{S}$
 $post : vida = \begin{cases} adev, & \text{dacă } s = \Phi \\ fals, & \text{dacă } s \neq \Phi \end{cases}$
- $plina(s)$
 $pre : s \in \mathcal{S}$
 $post : plina = \begin{cases} adev, & \text{dacă } s \text{ e plină} \\ fals, & \text{contrar} \end{cases}$
- $distruge(s)$
 {destructor}
 $pre : s \in \mathcal{S}$
 $post : s \text{ a fost 'distrusa' (spațiul de memorie alocat a fost eliberat)}$

Observații

- Stiva nu este potrivită pentru aplicațiile care necesită traversarea ei (nu avem acces direct la elementele din interiorul stivei).
- Afișarea conținutului stivei poate fi realizată folosind o stivă auxiliară (scoatem valorile din stivă punându-le pe o stivă auxiliară, după care se repun pe stiva inițială).

Implementări ale stivelor folosind

- tablouri - vectori (dinamici)
- liste înlănțuite.