

VECTOR

Vectorii sunt tablouri dinamice cu elemente omogene care au proprietatea de a se redimensiona automat când se adaugă sau se șterge un element, gestionarea memoriei fiind realizată automat de către container.

Vectorii sunt stocați într-o zonă contiguă de memorie și pot fi traversați cu ajutorul iteratorilor. Elementele se adaugă, de regulă la sfârșit. Operația de adăugare ia, de regulă, timp constant, cu excepția cazului când spațiul de memorie alocat trebuie redimensionat. Ștergerea unui element ia întotdeauna timp constant. Inserarea și ștergerea la începutul sau în interiorul vectorului iau timp liniar.

Pentru declararea unui vector trebuie inclusă biblioteca `<vector>` unde se află funcții de manipulare a acestui container.

```
vector<int>v;
```

Pentru rezervarea memoriei atunci când este cunoscut numărul de elemente se folosesc `reserve()` și `resize()`.

```
std::vector<int> vec1;  
vec1.reserve(30); // Se alocă spațiu pentru 30 de itemi, dar vec1 este încă gol.
```

```
std::vector<int> vec2;  
vec2.resize(30); // Se alocă spațiu pentru 30 de itemi, iar vec 2 conține acum 30 de itemi.
```

Dealocarea memoriei se face astfel:

```
v=vector<int>();
```

O altă metodă este prin utilizarea funcției `clear()` care elimină toate elementele containerului și setează dimensiunea acestuia la 0.

```
v.clear();
```

Pentru a elimina un anumit element din vector se folosește funcția `erase(i)` care elimină elementul de pe poziția `i`, iar dimensiunea acestuia se micșorează cu 1.

Vectorul nu este obligatoriu să fie gol el poate să fie declarat cu un anumit număr de poziții exact cum s-ar face la cei statici și să fie modificate pe parcurs. De asemenea el are în componență un constructor care spune cu ce să umple fiecare poziție.

`vector<int>v(100);` (vector cu 100 de elemente cu poziții de la 0-99 cu valoarea 0 implicită)

`vector<int>v(100,1);` (vector cu 100 de elemente cu poziții de la 0-99 cu valoarea 1)

- `push-back()` - inserează o valoare la sfârșitul vectorului.

Dacă se inserează o valoare într-un vector plin, dimensiunea acestuia crește automat(are loc o realocare automată a spațiului de stocare).

- `pop-back()` - este folosită pentru a elimina elemente de la sfârșitul vectorului

După ce ultima valoare din vector a fost eliminată, dimensiunea acestuia este redusă cu o poziție.

- `front()` - returnează o referință la primul element din vector
- `back()` - returnează o referință la ultimul element din vector
- `data()` – returnează un pointer direct la matricea de memorie utilizată intern de vector pentru a stoca elemente

Funcția `size()` returnează numărul de elemente aflate în vector, iar funcția `capacity()` returnează dimensiunea spațiului de stocare alocat în prezent vectorului. Relația dintre acestea este:

`capacity()>=size()`

Accesul la elementele vectorului se poate face fie cu operatorul [], fie cu metoda at(). Atât operatorul [i] cât și metoda at(i) returnează o referință la elementul de pe poziția 'i' din vector. Diferența este dată de faptul că dacă poziția accesată depășește dimensiunea vectorului, metoda at() returnează excepția **"out_of_range"**, în timp ce operatorul [] provoacă un comportament nedefinit.