Clase vector de la STL

I. Pita

Facultad de Informática - UCM

10 de octubre de 2017

Clase vector de la STL

- Secuencia de elementos de un cierto tipo `T`.
- Se declaran con un tamaño que aumenta automáticamente cuando se llenan.
- Cuando se necesita aumentar el tamaño el coste es lineal respecto al número de elementos que tenga el vector.
- Están definidos en la librería `<vector>`, bajo el espacio de nombres `std`.
- Para mas información sobre esta clase consultar:

`http://www.cplusplus.com/reference/vector/vector/`

Clase vector de la STL

- Al declarar una variable del tipo `vector`, hay que indicar el *tipo* de sus elementos.

```
std::vector<double> v; // vector con cero elementos
v.reserve(n); // reserva memoria
std::vector<double> w(n); // crea n posiciones
```

- La diferencia entre las dos declaraciones está en el número de elementos del vector después de la declaración.
  `v.size()` vale cero, y `w.size()` vale n
- Todas las componentes del vector `w` resultan inicializadas al valor por defecto del tipo.
- `n` puede ser el valor de una variable.

Clase vector de la STL. Tamaño...

El vector se utiliza a través de las funciones de la librería.

- `v.size()` devuelve el número de elementos que se han añadido al vector
- `v.capacity()` devuelve el número de elementos que puede almacenar el vector sin reservar nueva memoria.
- Acceso a los elementos

```
v[3] = 5; v.at[3] = 5;
x = v[0] + v[1]; x = v.at(0) + v.at(1);
```

La función at comprueba los límites del vector.

Clase vector de la STL. Tamaño...

- `v.emplace_back(...)` añade un elemento a la derecha del vector. Crea el objeto en su lugar del vector.
- `v.push_back(..)` añade un elemento a la derecha del vector. Crea el elemento y a continuación lo añade/mueve al vector.
- `v.clear()` elimina todos los elementos del vector, pero no la memoria reservada.
- `v.pop_back()` elimina el último elemento de un vector.

Clase vector de la STL. Recorridos

```
// Recorrido de un vector con iteradores implícitos
std::vector<char> vc;
// range-based loop
for (char c : vc)
  std::cout << c;

for (char& c : vc)
  std::cin >> c;

// Recorrido clásico
std::vector<char> vc;
for (size_t i = 0; i < vc.size(); ++i)
  std::cout << vc[i];
```

Clase vector de la STL. Búsquedas

```
// Búsqueda de un elemento
template <class T>
bool buscar (std::vector<T> const& v; T const& elem) {
  size_t i = 0;
  while (i < v.size() && v[i] != elem) ++i;
  return i < v.size();
}
//Si se quiere conocer la posición del elemento
template <class T>
bool buscar (std::vector<T> const& v; T const& elem,
size_t & pos)
{
  size_t i = 0;
  while (i < v.size() && v[i] != elem) ++i;
  if (i < v.size()) {pos = i; return true;}
  else return false;
}
```

Ordenar un vector

- Utilizar la función `sort` de la librería `algorithm`

```
// Ordenar todo el vector
    sort(v.begin(),v.end());

 // Ordenar los n primeros lementos
     sort(v.begin(),v.begin()+n);

// Ordenar los n últimos elementos
    sort(v.begin()+n,v.end());

// Ordenar los elementos entre las posiciones n y m
    sort(v.begin()+n,v.begin()+m);
```