

Contrôle de C++

Durée : 1h00

Aucun document autorisé

Nom :

Prénom :

Une *file d'attente* est une structure de données linéaire dans laquelle on entre par une extrémité et on en sort par l'autre. On définit les 4 opérations suivantes :

- **estVide** qui teste si la file est vide ou pas ;
- **premier** qui renvoie le premier élément de la file ;
- **defiler** qui enlève le premier élément de la file ;
- **enfiler** qui ajoute un élément à la fin de la file.

- 1. Écrivez la classe abstraite générique **File** avec les 4 opérations précédentes, la méthode **toString** et l'opérateur **<<** pour écrire dans un **std::ostream**.

- 2. Écrivez la classe **FileVideException** qui dérive de **std::exception**. Vous redéfinirez uniquement la méthode **what**.

-
- This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Les véhicules qui circulent en France sur les autoroutes sont classés en 5 catégories :

- classe 1 : véhicules légers (*e.g.* cabriolet) ;
- classe 2 : véhicules intermédiaires (*e.g.* camping-car) ;
- classe 3 : véhicules lourds à 2 essieux (*e.g.* autocar) ;
- classe 4 : véhicules lourds à +3 essieux (*e.g.* camion 38T) ;
- classe 5 : véhicules motos, side-cars...

- 4. Écrivez la classe **Vehicule** avec une variable membre protégée **classe** et une méthode **getClasse**.

- 5. Écrivez les classes **Cabriolet** et **Moto** avec leur constructeur. Quelle relation existe entre ces 2 classes et la classe **Vehicule** ?

- une variable privée `tarifs`, un tableau initialisé avec les tarifs pour les 5 catégories de véhicule;
- une méthode `getTarif` qui renvoie le tarif d'un véhicule en fonction de sa classe. Cette fonction a en paramètre un pointeur sur véhicule. **Attention : ne faire aucun test !**
- une méthode `encaisser` qui prend en paramètre un pointeur sur une `File` de pointeurs sur `Vehicule` et qui renvoie la somme totale des tarifs payés par les véhicules de la file.

-
- This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

-
- This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

-
-
-
-
-
-

Annexe

Member functions

(constructor)	constructs the vector (public member function)
(destructor)	destructs the vector (public member function)
operator=	assigns values to the container (public member function)
assign	assigns values to the container (public member function)
get_allocator	returns the associated allocator (public member function)

Element access

at	access specified element with bounds checking (public member function)
operator[]	access specified element (public member function)
front	access the first element (public member function)
back	access the last element (public member function)
data	direct access to the underlying array (public member function)

Iterators

begin cbegin (C++11)	returns an iterator to the beginning (public member function)
end cend (C++11)	returns an iterator to the end (public member function)
rbegin crbegin (C++11)	returns a reverse iterator to the beginning (public member function)
rend crend (C++11)	returns a reverse iterator to the end (public member function)

Capacity

empty	checks whether the container is empty (public member function)
size	returns the number of elements (public member function)
max_size	returns the maximum possible number of elements (public member function)
reserve	reserves storage (public member function)
capacity	returns the number of elements that can be held in currently allocated storage (public member function)
shrink_to_fit (C++11)	reduces memory usage by freeing unused memory (public member function)

Modifiers

clear	clears the contents (public member function)
insert	inserts elements (public member function)
emplace (C++11)	constructs element in-place (public member function)
erase	erases elements (public member function)
push_back	adds an element to the end (public member function)
emplace_back (C++11)	constructs an element in-place at the end (public member function)
pop_back	removes the last element (public member function)
resize	changes the number of elements stored (public member function)
swap	swaps the contents (public member function)