Vecteur générique [vc03] - Exercice

Karine Zampieri, Stéphane Rivière

Unisciel algoprog Version 21 mai 2018

Table des matières

Java - Vecteur générique (Solution)

Mots-Clés Structures vectorisées ■
Requis Axiomatique objet, Modèles – Programmation générique, Gestion des exceptions ■
Difficulté ●●○ (2 h) ■

Objectif
Cet exercice réalise une classe générique `Vecteur` qui modélise les conteneurs vectorisées d'éléments de type `T`.

1 Classe Vecteur<T>

1.1 La classe Vecteur

Similairement au modèle de classes `Vector<T>`, trois attributs seront créés :

- Un pointeur `arr` sur le tableau dynamique des éléments.
- Un entier `capacity` traduisant la capacité de stockage physique du vecteur.
- Un entier `size` mémorisant le nombre d'éléments du vecteur.

Écrivez une classe générique `Vecteur<T>` qui contient ces trois attributs.

1.2 Les structeurs

Écrivez une méthode **interne** `x_reserve(n,copy)` qui étend la capacité du vecteur pour `n` éléments et copie les éléments existants dans le nouvel espace si le booléen `copy` est `Vrai`.

Écrivez une méthode **interne** `x_copy(obj)` qui copie les éléments d'un `Vecteur<T>` `obj` dans l'objet courant.

Écrivez le constructeur par défaut.

Écrivez le constructeur issu de `n` éléments.

Écrivez une méthode `clear` qui purge le conteneur.

Écrivez le constructeur de recopie.

Écrivez une méthode `clone` qui renvoie la copie de l'objet courant.

1.3 Les accesseurs

Écrivez l'accesseur `size` de la taille.

Écrivez le prédicat `isEmpty` de conteneur vide.

Écrivez l'accesseur `capacity` de la capacité.

Écrivez l'accesseur `get(k)` de la référence au `k`-ème élément ainsi que le modifieur `set(k,e)` qui fixe la nouvelle valeur `e` au `k`-ème élément.

Écrivez l'accesseur contrôlé `at(k)` de la référence au `k`-ème élément.
Lancez une exception si l'indice `k` n'est pas valide.

Écrivez l'accesseur `lastElement` de la référence au dernier élément.
Lancez une exception si le conteneur est vide.

1.4 Les méthodes

Écrivez une méthode `add(e)` qui insère une valeur `T e` en queue du conteneur.

Écrivez une méthode `add(k,e)` qui insère une valeur `T e` en position `k` dans le conteneur.
Lancez une exception si le conteneur est vide ou si `k` n'est pas valide.

Écrivez une méthode `remove(k)` qui supprime l'élément en position `k` du conteneur.
Lancez une exception si le conteneur est vide ou si `k` n'est pas valide.

Validez votre classe avec la solution.

Solution Java @[Vecteur.java]

```java
import java.lang.RuntimeException;

class ErreurIndexRange extends Exception{
   public ErreurIndexRange(String s,int k,int n){
      super(s);
      System.err.println(" k="+k+" n="+n);
   }
}

public class Vecteur<T>{
private int m_capacity;
private int m_size;
private T[] m_arr;

private void x_reserve(int n,boolean copy){
   T[] newArr = (T[])new Object[n];
   if (copy){
      for (int k = 0; k < m_size; ++k){
         newArr[k] = m_arr[k];
      }
   }
   m_arr = newArr;
   m_capacity = n;
}

private void x_copy(Vecteur<T> obj){
   m_size = obj.size();
   for (int k = 0; k < m_size; ++k){
      m_arr[k] = obj.m_arr[k];
```

```
    }
}

public Vecteur(){
    m_size = 0;
    m_capacity = 0;
    m_arr = null;
}

public Vecteur(int n){
    m_size = 0;
    m_capacity = 0;
    m_arr = null;
    if (n != 0){
        x_reserve(n,false);
        m_size = n;
        for (int k = 0; k < m_size; ++k){
            m_arr[k] = null;
        }
    }
}

public void clear(){
    m_size = 0;
    m_capacity = 0;
    m_arr = null;
}

public Vecteur(Vecteur<T> obj){
    m_size = 0;
    m_capacity = 0;
    m_arr = null;
    if (obj.size() != 0){
        x_reserve(obj.size(),false);
        x_copy(obj);
    }
}

public Vecteur<T> clone(){
    Vecteur<T> obj = new Vecteur<T>(this.size());
    obj.x_copy(this);
    return obj;
}

public int size(){
    return m_size;
}

public boolean isEmpty(){
    return m_size == 0;
}

public int capacity(){
    return m_capacity;
}

public T get(int k){
```

```
   return m_arr[k];
}

public T set(int k,T e){
   m_arr[k] = e;
   return e;
}

public T lastElement(){
   if (isEmpty()){
      throw new RuntimeException("Vecteur lastElement(): vecteur vide");
   }
   return m_arr[m_size-1];
}

public boolean add(T e){
   if (size() == capacity()){
      if (capacity() == 0){
         x_reserve(1,false);
      }
      else{
         x_reserve(2*capacity(),true);
      }
   }
   m_arr[m_size] = e;
   m_size += 1;
   return true;
}

public void add(int k,T e) throws RuntimeException, ErreurIndexRange{
   if (isEmpty()){
      throw new RuntimeException("Vecteur insert(): vecteur vide");
   }
   if (k < 0 || k > size()){
      throw new ErreurIndexRange("Vecteur insert(): ErreurIndexRange",k,size());
   }
   if (size() == capacity()){
      x_reserve(2*capacity(),true);
   }
   for (int j = size()-1; j >= k; --j){
      m_arr[j+1] = m_arr[j];
   }
   m_arr[k] = e;
   m_size += 1;
}

public void remove(int k)throws RuntimeException, ErreurIndexRange{
   if (isEmpty()){
      throw new RuntimeException("Vecteur erase(): vecteur vide");
   }
   if (k < 0 || k > size()){
      throw new ErreurIndexRange("Vecteur erase(): ErreurIndexRange",k,size());
   }
   for (int j = k+1; j < size(); ++j){
      m_arr[j-1] = m_arr[j];
   }
   m_size -= 1;
}
```

```
}
```

1.5 Programme de test

Écrivez une procédure `test_vecteur` qui teste votre classe.

Testez.

Écrivez une procédure `test_exceptions` qui teste les exceptions.

Testez.

Validez vos procédures avec la solution.

Solution Java @[pgvecteur.java]

```java
public class PGVecteur {
static void afficherV(Vecteur<Integer> v){
   for (int j = 0; j < v.size(); ++j){
      System.out.print(v.get(j)+" ");
   }
   System.out.println();
}

static void test_vecteur(){
   Vecteur<Integer> v = new Vecteur<Integer>();
   for (int j = 0; j < 5; ++j){
      v.add(j*2+(j%2));
   }
   afficherV(v);
   try{
      v.remove(2);
      afficherV(v);
      v.add(0,-2);
      afficherV(v);
   }
   catch (Exception e){
      e.printStackTrace();
      System.err.println("OUPS "+e);
   }
}

public static void main(String[] args) {
   test_vecteur();
}
}
```

2 Références générales

Comprend [] ■