

<b>Classe : INF2</b>	<b>Labo no : 04</b>	Exceptions
----------------------	---------------------	------------

## But

- Classes génériques Vecteur et Matrice
- Utilisation des exceptions

## A faire

Développer dans des fichiers séparés distincts deux classes génériques « Vecteur » (`vector<T>`) et « Matrice » (`Vecteur<T>`) permettant la gestion de données quelconques.

Fonctionnalités demandées

### Vecteur

<code>Vecteur(n)</code>	Constructeur avec la taille $n$ initiale
<code>Vecteur(vector&lt;T&gt;)</code>	Constructeur sur la base d'un <i>vector</i>
opérateur <code>&lt;&lt;</code>	affiche le contenu au format <code>[a, b, ...]</code>
<code>at(n)</code>	Accès à l'élément en position $n$ (lecteur et écriture)
<code>size()</code>	Retourne la taille actuelle
<code>resize(taille)</code>	Redimensionne le vecteur
<code>somme()</code>	Retourne la somme des éléments contenus
Opérateur <code>*</code> (valeur)	Multiplie chaque élément par <i>valeur</i> et retour le vecteur correspondant ex : $v * [a, b, c] \Rightarrow [v*a, v*b, v*c]$
Opérateur <code>*</code> (vecteur)	Multiplie chaque élément entre eux et retourne le vecteur correspondant $[a, b, c] * [d, e, f] \Rightarrow [a*d, b*e, c*f]$
Opérateur <code>+</code> (vecteur)	Retourne un vecteur correspondant à l'addition de chacun des éléments ex : $[a, b, c] + [d, e, f] \Rightarrow [a+d, b+e, c+f]$
Opérateur <code>-</code> (vecteur)	Retourne un vecteur correspondant à la soustraction de chacun des éléments ex : $[a, b, c] - [d, e, f] \Rightarrow [a-d, b-e, c-f]$

## Matrice

Matrice()	Constructeur par défaut
Matrice(l)	Matrice de $l$ lignes et 0 colonne
Matrice(l, c)	Matrice de $l$ lignes et $c$ colonnes
operateur <<	affiche le contenu au format $[[a, b][c, d], \dots]$
at(n)	Accès au vecteur en position $n$ (lecteur et écriture)
size()	Retourne la taille actuelle (nbre de lignes)
resize(l)	Redimensionne le vecteur à $l$ lignes
resize(l, c)	Redimensionne le vecteur à $l$ lignes et $c$ colonnes
estVide()	Indique que la matrice ne contient aucune valeur
estCarree()	Indique que la matrice est carrée. Par convention une matrice vide est carrée
estReguliere()	Indique que toutes les lignes ont la même longueur. Par convention une matrice vide est régulière
sommeLigne()	Retourne un vecteur contenant la somme de chaque ligne
sommeColonne()	Retourne un vecteur contenant la somme de chaque colonne
sommeDiagonaleGD()	Retourne la somme des valeurs de la diagonale Gauche-Droite
sommeDiagonaleDG()	Retourne la somme des valeurs de la diagonale Droite-Gauche
Operateur * (valeur)	Retourne la matrice dont tous les éléments ont été multipliés par <i>valeur</i> . Ex : $n * [[a], [b, c]] \Rightarrow [[n*a], [n*b, n*c]]$
Operateur * (matrice)	Retourne une matrice dont tous les éléments ont été multipliés par les éléments correspondants. L'opération est possible seulement si les matrices ont strictement la même forme. Ex : $[[a], [b, c]] * [[g], [h, i]] = [[a*g], [b*h, c*i]]$
Operateur + (matrice)	Retourne une matrice dont tous les éléments ont été additionnés avec les éléments correspondants. L'opération est possible seulement si les matrices ont strictement la même forme. Ex : $[[a], [b, c]] + [[g], [h, i]] = [[a+g], [b+h, c+i]]$

# L a b o r a t o i r e

Intégrer dans ce développement des exceptions qui seront traitées par l'appelant (main) de sorte à fournir un message parlant du problème rencontré.

Pour vous aider, consulter les exceptions liées aux méthodes des *vector* que vous utilisez pour déterminer ce qui pourrait se passer. En plus d'un message clair, les messages d'exception doivent être pré-fixés « Vecteur » ou « Matrice » selon la provenance.

Créer un programme de test afin de démontrer le bon fonctionnement de ces deux classes. Les tests des cas particuliers selon, au besoin, en commentaire.