

Classe : INF2	Labo no : 04	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

Vecteur(n)	Constructeur avec la taille n initiale
Vecteur(vector<T>)	Constructeur sur la base d'un <i>vector</i>
opérateur <<	affiche le contenu au format [a, b, ...]
at(n)	Accès à l'élément en position n (lecteur et écriture)
size()	Retourne la taille actuelle
resize(taille)	Redimensionne le vecteur
somme	Retourne la somme des éléments contenus
Opérateur * (valeur)	Produit par une valeur
Opérateur * (vecteur)	Produit par un autre vecteur
Opérateur + (vecteur)	Addition d'un autre vecteur
Opérateur - (vecteur)	Soustraction d'un autre vecteur

Matrice

Matrice()	Constructeur par défaut
Matrice(l)	Matrice de l lignes et 0 colonne
Matrice(l, c)	Matrice de l lignes et c colonnes
opérateur <<	affiche le contenu au format <code>[[a, b][c, d], ...]</code>
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
Opérateur * (valeur)	Produit par une valeur
Opérateur * (matrice)	Produit par une autre matrice
Opérateur + (matrice)	Addition d'une autre matrice

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.