Cours 420-202-RE **Traitement de données orienté objet** **Hiver 2021** **Cégep Limoilou** **Département d'Informatique**	**Tp 5 mandat 2 de 4 (1 semaine pour cette partie)** **12% pour les 4 mandats** **Cryptographie - Le chiffre de Hill**

OBJECTIFS

- Utiliser et manipuler des structures de données sur disque et en mémoire;
- Trouver une solution informatique à un problème;
- Utiliser les méthodes appropriées des classes de l'API de Java;
- Livrer un code documenté et testé.

ACTIVITÉ À RÉALISER

- Voir le fichier du mandat 1.

MANDAT 2 :

Documentation et recherche :

- Voir le fichier du mandat 1.
- Voir les notes de cours et les exercices sur la récursivité.

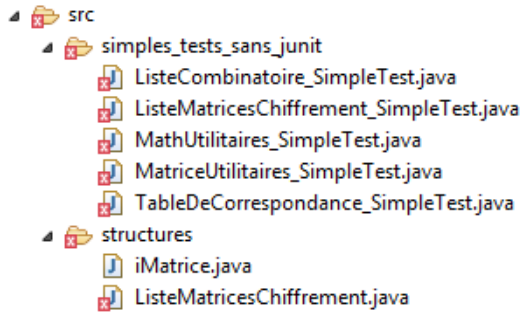
Gestion des matrices de chiffrement et de déchiffrement :

- **Très important** : Avant de commencer le mandat 2, il faut terminer et bien tester tous les éléments du mandat 1.
- Commencer le mandat 2 en développant les méthodes qui comportent dans leur notice « TODO » la référence « MANDAT 2 » disponible à partir du code du mandat 1. Adaptez vos classes de tests JUnit en conséquence.
- Maintenant que vous avez avancé dans le développement des utilitaires et structures de données qui forment un « framework » minimal pour la suite de votre laboratoire, vous pouvez copier le code du mandat 2, disponible sur le réseau, et l'intégrer à votre projet du mandat 1.
- À partir du code fourni sur le réseau vous devez, selon la « JavaDoc » complétez les différentes méthodes de la classe « ListeMatricesChiffrement ».
    - Dans le package « structures » vous avez la nouvelle classe « ListeMatricesChiffrement » qui implémente l'interface « iMatrice » et qui gère une liste de matrices candidates pour le chiffrement de Hill selon une dimension de matrice et un coefficient dans Z voulus.

      Attention : une dimension de 3 donne des combinaisons de 9 et chaque combinaison sera utilisée comme valeurs dans une matrice.

      La classe « ListeMatricesChiffrement » doit à partir d'une liste combinatoire, trouver et stocker les matrices candidates. Elle doit également permettre de choisir une matrice parmi celles dans la liste comme matrice de chiffrement (matrice courante) et nous donner, à la demande, son inverse dans Hill, soit la matrice de déchiffrement. (voir plus bas)
    - Il faut aussi produire une classe de tests JUnit pour la classe que vous venez de compléter dans ce mandat. Il est conseillé de produire les tests au fur et à mesure que vous développez vos méthodes. **Testez correctement car ça fait toujours partie de la base de votre solution globale, vous devez avoir confiance en votre code.**

- À partir du code fourni sur le réseau vous trouverez aussi, dans le package « simples_tests_sans_junit », un ensemble de classes qui utilisent de <u>façon sommaire</u> le « framework » que vous développez (mandat 1 et 2). J'ai mis en commentaires les sorties que j'ai obtenues à partir de mon code et des valeurs d'origines qui sont dans les différentes classes de tests. **<u>Ces tests sont simplement fournis pour valider vos sorties avec les miennes. Il ne faut pas considérer ces tests comme étant suffisants pour dire que tout fonctionne correctement.</u>**
- Voici les différents packages et classes qui vous sont fournis pour ce mandat.



Échéancier :

- Vous avez une semaine pour réaliser le travail demandé par ce mandat.
- Il n'y a rien à remettre pour ce mandat, vous devez simplement avoir terminé et testé correctement votre code avant de recevoir le prochain mandat.

Matrice inverse de Hill :

1. Trouver la matrice adjointe
2. Trouver le déterminant inverse de Hill
    a. Trouver l'ensemble des nombres qui sont premiers entre eux avec 0 et force brute (coef Z ex. 28)
    b. Pour chaque nb premier
        i. Multiplier déterminant matrice par le nb
        ii. Si le produit précédent modulo force brute = 1 → C'est le déterminant inverse
3. Faire le produit scalaire de la matrice adjointe par le déterminant inverse
4. Faire le modulo de matrice produit scalaire (calculée en 3) et force brute (coef Z)

Exemple :

Matrice choisie :

1	3	4
5	6	7
8	9	10

Déterminant = 3

1) Matrice adjointe :

-3	6	-3
6	-22	13
-3	15	-9

2) x premiers nombres entre 0 et 28 : 1, 3, 5, 9, 11, 13, 15, 17, 19, 23, 25 27

Essayer avec chacun des 12 nombres premiers lequel multiplié par le déterminant modulo 28 donne 1
Ici : 19 * 3 = 57 modulo 28 = 1. Donc le déterminant est 19

3) Matrice adjointe * déterminant inverse

-3	6	-3
6	-22	13
-3	15	-9

* 19 -→

-57	114	-57
114	-418	247
-57	285	-171

4) Matrice résultant du produit calculé en 3) modulo 28

-57	114	-57
114	-418	247
-57	285	-171

Mod 28 →

27	2	27
2	2	23
27	5	25

Matrice inverse de Hill