

## Série H : Exceptions et fichiers

### Table des matières

Exercice H.1 : CryptoBox .....	2
--------------------------------	---

## Exercice H.1 : CryptoBox

Téléchargez le projet **CryptoBox** de eduMoodle. Il s'agit de développer une application pour crypter/décrypter un message.



Pour crypter un message, les codes ASCII de chaque caractère du message sont inscrits colonne par colonne dans une matrice M, composée de 3 lignes. En multipliant M par la matrice C qui constitue la « clé de cryptage » on obtient le message chiffré sous forme d'une matrice B. On a donc  $C \cdot M = B$ . Parcourez la matrice B colonne par colonne pour former le message crypté où les différents nombres sont séparés par un point-virgule.

Pour décrypter un message il faut réaliser l'opération inverse, sachant que  $M = C^{-1} \cdot B$ .

### Travail à faire :

- Implémentez la méthode **public int[][] multiply(int[][] A, int[][] B)** qui retourne le produit de deux matrices A et B.
- Implémentez la méthode **public String encodeMessage(String textToEncode)** qui retourne le message crypté sous forme de chaîne de caractères.
- Implémentez la méthode **public String decodeMessage(String encodedMessage)** qui retourne le message déchiffré sous forme de chaîne de caractères.
- Ajoutez la possibilité de charger/sauvegarder des messages cryptés. Ces fichiers portent l'extension « **.cry** » et l'emplacement et le nom de ces fichiers est sélectionné par des menus de chargement/sauvegarde.
- Déchiffrer le message suivant :

932;343;-138;562;198;-45;936;351;-120;1090;402;-185;828;309;-106;465;193;-46

### Pour les pros :

Calculez l'inverse de la Matrice C automatiquement. Si ce n'est pas possible, jetez une exception de type *ArithmeticError*.