



## TD Gestion d'erreur et Javadoc

### Résumé

Ces exercices ont pour but de vérifier que vous avez fixé comment découper un algorithme, comment lancer une exception et documenter votre code.

|          |                        |          |
|----------|------------------------|----------|
| <b>1</b> | <b>Découpe du code</b> | <b>2</b> |
| 1.1      | À vous de jouer...     | 2        |



# 1 Découpe du code

## 1.1 À vous de jouer...

Voici quelques conseils pour vous guider dans la résolution de tels problèmes :

- il convient d’abord de bien comprendre le problème posé ; assurez-vous qu’il est parfaitement spécifié ;
- résolvez le problème via quelques exemples précis ;
- mettez en évidence les variables **«données »**, les variables **«résultats »** et les variables de travail ;
- n’hésitez pas à faire une ébauche de résolution en français avant d’élaborer l’algorithme définitif pseudo-codé ;
- déclarez ensuite les variables (et leur type) qui interviennent dans chaque algorithme ; les noms des variables risquant de ne pas être suffisamment explicites.
- Écrivez la partie algorithmique **AVANT** de vous lancer dans la programmation en Java.

### What time is it ?

Le système horaire anglais est basé sur un cycle de 12 heures («12-hour clock ») au lieu du système continental utilisant un cycle de 24 heures. Pour distinguer les moments de l’avant-midi et de l’après-midi, on utilise les abréviations AM (ante-meridiem) pour tout moment entre minuit et midi et PM (post-meridiem) pour tout moment entre midi et minuit. Les heures sont représentées par les nombres de 1 à 12 (et jamais par 0, qui n’est pas utilisé pour les heures). Il y a 2 cas spéciaux : comme midi ne peut être avant ou après lui-même, son écriture correcte dans le système anglais est «12 noon », de même que minuit s’écrit «12 midnight ». Il est donc incorrect d’écrire 12 AM ou 12 PM.

Mettez en évidence les variables **«données »**, les variables **«résultats »** et les variables de travail ;

Écrivez un algorithme qui reçoit en paramètre des heures et des minutes et affiche l’heure correspondante au format anglais. (Dans cet exercice, on ne tient donc pas compte des secondes.)

Exemples :

- 0h18 devient 12 :18 AM
- 8h15 devient 8 : 15 AM
- 12h00 devient 12 noon
- 12h30 devient 12 : 30 PM
- 13h25 devient 1 : 25 PM

- 23h42 devient 11 :42 PM
- 0h00 devient 12 midnight

Si les nombres entrés ne sont pas dans les bornes acceptées pour les heures et les minutes, vous lancerez une erreur qui arrêtera le programme.

Écrivez le code java correspondant avec la documentation javadoc pour chaque méthode. N'oubliez pas d'écrire la méthode `main` qui fera appel à votre méthode.

### La plus grande somme

Écrire un algorithme qui reçoit deux nombres entiers positifs de 5 chiffres et retourne celui dont la somme des chiffres est la plus grande. Mettez en œuvre tout ce qu'il faut pour ne pas devoir répéter plusieurs fois le même code !

Si les nombres entrés ne sont pas positifs, vous lancerez une erreur qui arrêtera le programme.

Par exemple, si les nombres entrés sont 11789 et 50211, l'algorithme retourne 11789.

Mettez en évidence les variables «**données** », les variables «**résultats** » et les variables de travail ;

Écrivez le code java correspondant avec la documentation javadoc pour chaque méthode. N'oubliez pas d'écrire la méthode `main` qui fera appel à votre méthode.

### La boule de cristal

Cet algorithme est destiné à prédire l'avenir, et il doit être infallible ! Il recevra un moment de la journée en paramètre (sous forme de trois entiers : heure, minute et seconde) et affichera le moment qu'il sera une minute plus tard !

Par exemple, si le moment entré est 21h 32' 45", l'algorithme affichera 21h 33' 45".

Si les nombres entrés ne sont pas dans les bornes acceptées pour les heures, les minutes et les secondes, vous lancerez une erreur qui arrêtera le programme.

Si vous avez déjà fait ce test pour les heures et les minutes, avez-vous pensé à modulariser ? Si pas, c'est le moment de le faire :)

Mettez en évidence les variables «**données** », les variables «**résultats** » et les variables de travail ;

Écrivez le code java correspondant avec la documentation javadoc pour chaque méthode. N'oubliez pas d'écrire la méthode `main` qui fera appel à votre méthode.

Pour plus d'exercices, révissez ici ([www.heb.be/esi/InitModule/fr/../../../../TModule/fr/html/Exercices\\_learningObject3.html](http://www.heb.be/esi/InitModule/fr/../../../../TModule/fr/html/Exercices_learningObject3.html))