

SAE 1.02 | équipe 22 | partie 1 - Exercice 2

LEPILLER	Floriane	A2
LETELLIER	Axel	A2
KULPA	Clément	B2
HERMILLY	Joshua	C2

Exercice 2

1) Dans votre compte-rendu, faites les deux propositions de structures en expliquant les impacts sur leur manipulation pour pouvoir faire fonctionner notre horloge.

Pour faire fonctionner notre horloge nous avons 2 solutions:

La première consiste à stocker le temps convertit en seconde dans une seule variable. Cette méthode est plus compacte et réduite en place. Par ailleurs, il nous suffira de manipuler une seule variable et ainsi réduire certaines erreurs.

La seconde vise à séparer les valeurs dans des variables différentes. Cette méthode permettra un langage plus compréhensible et ainsi une lecture plus aisée. Cependant, l'utilisation de plusieurs variables est plus propice à limiter les erreurs de conversion.

2) Écrire deux programmes permettant de comparer les temps d'exécution de ces deux formes de stockage.

Notre programme *HorlogeSeconde.java* stock le temps totalement en secondes, et le programme *Horloge.java* en Heures, minutes, secondes.

3) Dans votre compte-rendu, Vous consignerez votre protocole de tests, vos résultats, ainsi que l'analyse de ces résultats. Vous joindrez également les .java de vos deux programmes.

Les résultats de notre algorithme *HorlogeNormal* :

```
hj243638@c-di-727-16:~/TP/s1/r1.01_init_dev$ java HorlogeNormal
Temps : 1.41990579
```

```
hj243638@c-di-727-16:~/TP/s1/r1.01_init_dev$ java HorlogeNormal
Temps : 1.41138902
```

```
hj243638@c-di-727-16:~/TP/s1/r1.01_init_dev$ java HorlogeNormal
Temps : 1.434762987
```

Les résultats de notre algorithme *HorlogeSeconde* :

```
hj243638@c-di-727-16:~/TP/s1/r1.01_init_dev$ java HorlogeSeconde  
Temps : 1.386685728
```

```
hj243638@c-di-727-16:~/TP/s1/r1.01_init_dev$ java HorlogeSeconde  
Temps : 1.387492135
```

```
hj243638@c-di-727-16:~/TP/s1/r1.01_init_dev$ java HorlogeSeconde  
Temps : 1.418607346
```

Grâce à ces résultats nous voyons donc que notre programme qui n'utilise qu'une variable seconde est plus rapide.

A chaque itération nous obtenons une différence d'environ 0.1 seconde. Cela s'explique par le fait que dans le programme *HorlogeNormal* exécute des calculs et tests supplémentaires comparé à l'algorithme *HorlogeSeconde*.