

# SAE 1.02 Comparaison d'approches algorithmiques

## Exercice n°1

- 1) byte : Désigne un octet (permet de coder un entier compris entre -128 et 127)  
short : C'est un entier avec une taille de 2 octets (compris entre  $-2^{15}$  à  $(2^{15}) - 1$  )  
int : Désigne un entier codé sur 4 octets (compris entre  $-2^{31}$  et  $(2^{31}) - 1$  )  
long : Désigne un entier codé sur 8 octets (compris entre  $-2^{63}$  et  $(2^{63}) - 1$  )
- 2) On suppose que ces types ont une différence minime en termes de performances.  
Mais malgré tout existante, probablement dépendante de la plage de valeur. Plus elle est petite, plus elle sera rapide mais cela s'exprime probablement en nanoseconde.

En toute logique, si nous devons les classer du plus performant au moins performant, nous aurions:

- 1) byte
  - 2) short
  - 3) int
  - 4) long
- 3) En toute logique, d'après notre réponse précédente, une boucle devrait également prendre une différence de temps en fonction du type utilisé.

Pour comparer dans les mêmes conditions les types short, int et long, nous devrions créer une boucle while qui boucle avec le type en question, reproduire cette boucle pour chaque type que l'on souhaite comparer et calculer le temps d'exécution pour chacun.

- 4) Lors de l'exécution, le long semble plus performant que le type short en termes de durée. Cependant, le short est moins performant que le type int

le classement est donc :

- 1) int
- 2) long
- 3) short

Résultat avec le short :

```
PS C:\Users\moonr\OneDrive\Documents\python> java BoucleShort  
Temps: 48535300 nanosecondes
```

Résultat avec le int :

```
PS C:\Users\moonr\OneDrive\Documents\python> java BoucleInt  
Temps: 1380300 nanosecondes
```

Résultat avec le long :

```
PS C:\Users\moonr\OneDrive\Documents\python> java BoucleLong  
Temps: 2717900 nanosecondes
```

Nous avons fait plusieurs tests et nous obtenons régulièrement des résultats similaires à ceux ci-dessus