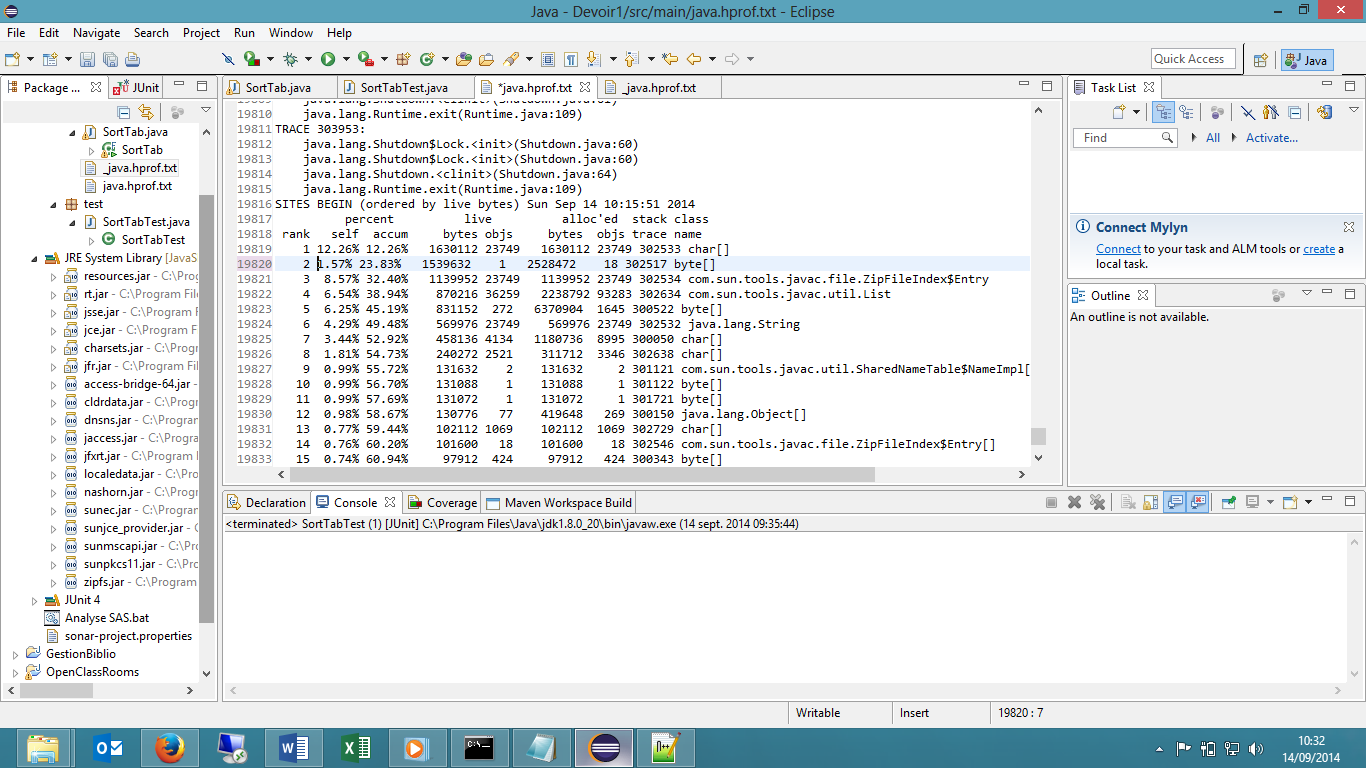
1. *Profilage en Mémoire des Classe SortTab() et SortTabTest()*

L’observation du rapport généré montre que :

* 23 749 objets ont été créés pour 23 749 objets en mémoire de type char []
* 18 objets ont été créés pour 1 objet en mémoire de type byte [], les variables des tableaux étant déclarés de type Integer, il serait judicieux de les conserver comme tel.

Vu l’espace dont on dispose, le Garbage collector n’est pas utiliser inutilement, puis le temps n’est pas préjudiciable.



1. *Profilage en Mémoire des Classe SortTab() et SortTabTest()*

L’observation du rapport généré montre que les goulots d’étranglement ou les méthodes les plus consommatrices en temps sont « **java.lang.Object.wait appelé 17 fois** » et « **java.lang.ref.ReferenceQueue.remove appelé 782 fois** ». Ce sont des threads natifs demandant un approfondissement pour les résoudre.

