|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Temps | ArrayList | LinkedList | Tableau | Array/Linked | Array/Tableau | Linked/Tableau |
| 1 seconde | 10492024 | 5465514 | 14036493 | 1,919677 | 0,747482 | 0,389379 |
| 2 secondes | 22170157 | 10498469 | 29711619 | 2,111751 | 0,746178 | 0,353346 |
| 5 secondes | 56667155 | 23610420 | 72576806 | 2,400091 | 0,780789 | 0,325316 |
| 8 secondes | 87666239 | 35953602 | 1,18E+08 | 2,438316 | 0,742853 | 0,304658 |
| 10 secondes | 113146434 | 47595318 | 1,39E+08 | 2,37726 | 0,811656 | 0,341425 |

Conclusion :

D’après les résultats obtenus par la mesure de performances ***de ArrayList, LinkedList et le tableau*** simple, nous avons pu collecter les données dans le tableau ci-dessus.  
Nous avons pu constater que durant notre expérience, que pour tout d’abord les « ArrayList », on peut voir qu’en 10 secondes par exemple, ce un ArrayList peut additionner plus d’éléments que « LinkedList » mais par contre le tableau simple possède plus d’éléments en 10 secondes que ArrayList.

Donc après cette expérience, on peut conclure qu’à des fins d’optimisation d’un programme, le mieux serait d’utiliser un tableau simple car la machine dépense moins de ressources pour créer un tableau simple.