Détermination de X à partir d’une méthode d’erreur :

Afin de déterminer le X (la valeur suffisamment grande pour laquelle il faut plus de 10s au tableau pour se remplir), la méthode d’erreur prend en paramètre la taille du tableau à tester et mesure le temps qu’elle prend pour faire l’addition des valeurs dans le tableau . Si le temps est inferieur est inférieur à 10 secondes, la fonction retourne false sinon elle retourne vraie.

20 000 000 est la valeur que nous avons retenue

**public** **static** **Boolean** arraylistFillRuntime(**int** sizeArray ) // recuperer avec sizeArray la taille a tester

{

List <Character> tableau = **new** ArrayList<Character>();// creation d’un tableau dynamique Arraylist

**double** heure = (**double**) (System.*currentTimeMillis*()/1000); // Mesure du temps initial

// remplir le tableau

**for** (**int** i=0; i< sizeArray; i++)

{

tableau.add(*random\_char*());

}

//prendre l'heure a la fin //determiner le temps que ca a pris

**double** heure\_fin = (**double**) (System.*currentTimeMillis*()/1000);

heure = heure\_fin - heure ;

// comparer le temps ecoule avec 10s ;

**if** (heure < 10)

**return** **false** ;

**else**

**return** **true**;

}

Résultats des comparaisons

Afin de réaliser les comparaisons, nous avons remplis tous les tableaux (Arrays, Arraylist, Linkedlist) de 20 000 000 éléments et avons comparer le temps écoulé pour chacun des structures de données.

Le résultat se présente comme suit :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre d'éléments | ArrayList | LinkedList | Tableau | Array/Linked | Array/Tableau | Linked/Tableau |
| 10 000 000 | 756 | 1628 | 459 | 0,464373 | 1,647059 | 3,546841 |
| 25 000 000 | 1618 | 4736 | 1118 | 0,341639 | 1,447227 | 4,236136 |
| 50 000 000 | 3235 | 9677 | 2275 | 0,334298 | 1,421978 | 4,253626 |
| 75 000 000 | 5110 | 14711 | 3687 | 0,347359 | 1,385951 | 3,989965 |
| 100 000 000 | 6565 | 19843 | 4330 | 0,330847 | 1,516166 | 4,582679 |

Nous constatons que l’addition des éléments se fait plus vite lorsqu’on utilise des tableaux simples. Par conséquent, un concepteur qui connait déjà la plage de données avec laquelle il doit travailler aurait tout intérêt à utiliser un tableau pour stocker ses données.

Cependant si, pour un besoin de flexibilité, un tableau dynamique est le meilleur choix , alors la structure de données à privilégier est la ArrayList.