

Programme java TabCouple

Notre programme se divise en deux classes Couple et TabCouple

Le choix de l'implémentation :

Dans ce TP il nous fallait créer une classe simulant un tableau dynamique de Couple géré de manière transparente pour l'utilisateur. Ainsi il a fallu faire la distinction entre la taille du tableau et le nombre d'éléments présents dans celui-ci. La variable `length` indique la taille du tableau et la variable `index` indique le nombre d'éléments qui sont présents dans le tableau. Elle indique aussi l'endroit où insérer un Couple lorsque l'utilisateur veut en rajouter un.

Il y a deux constructeurs

- * `TabCouple()` --> la variable `size` = 10 par défaut
- * `TabCouple(int)` --> constructeur par défaut + définit la variable `size`

Les fonctions présentes permettent l'ajout/suppression d'un Couple dans le tableau. La sélection/remplacement d'un Couple dans le tableau

- * `add(Couple)` --> ajoute le couple en dernière position
- * `remove()` --> retire le dernier couple du tableau
- * `get(int)` --> retourne le couple placé en position `i`
- * `set(int, Couple)` --> remplace le Couple en position `i` par celui passé en paramètre

Pour les fonctions privées, la classe dispose de :

- * `grow(int size)` --> agrandi le tableau de `size` slot
- * `grow()` --> `grow(size)`
- * `shrink()` --> réduit la taille du tableau de moitié en laissant tout de même «size» places de libre

Le principe est que lorsque l'utilisateur rajoute un Couple dans le tableau alors que celui-ci est plein, le tableau s'agrandi pour pouvoir accueillir size Couple de plus. De même lorsque le nombre d'éléments du tableau devient trop faible par rapport à sa taille, celui-ci est réduit de moitié ($2 * \text{index} < \text{length}$).

Note : la réduction du nombre de slot n'entraîne jamais la perte de Couples présents dans le tableau.

Pour le tri par tas :

On tri le tableau à la façon d'un arbre binaire tassé (cf. le pere en i est plus grand que ses 2 fils en $2*i$ et $2*i+1$). Puis, en plaçant la racine en fin de tableau et en retassant le dernier élément de l'arbre on obtient le tableau trié final.