ANNEXE 17 - Les collections – Stack , Vector , List, etc.

## Les Vectors

* Une Stack est une mise en œuvre simple d’une classe plus générale : Vector
* Un Vector vise à représenter un tableau d’objets, chaque item du tableau pouvant être accessible à partir de son index dans le tableau.
* Cependant, la capacité d’un Vector peut s’agrandir au fur et à mesure qu’on ajoute des items dans le Vector 🡪 allocation dynamique de mémoire
* Attention, il ne peut pas avoir d'éléments vides entre des éléments occupés à l'intérieur du Vector

EX. :

***Constructeurs :***

Vector premier = new Vector(); Quand tu crée un vecteur il y a de la mémoire aloué de 10 place donc de 0 …. 9

Vector deuxieme = new Vector(2);Crée capacité de vecteur avec une capacité initiale de 2 place donc 0 1

Vector<Double> troisieme = new Vector<Double>(3);Cliquez ou appuyez ici pour entrer du texte.

***Méthodes principales :***

void add ( Object o ) : Ajoute l objet o a la premiere place dispo dans le vecteur

void insertElementAt ( Object o, int index ) : si on veut inserer un objet dans une place precise faut ajouter le int et ensuite l index. Et pousse les element suivant si necessaire

Object get ( int index ) : retourne une reference a l objet present a cet index

Object remove ( int index): retourne une refernce de l objet et le retire du vecteur

boolean remove ( Object o ): retourne la premiere occurrence de l objet passé en parametre

boolean contains (Object o ): pour vérifier si un element est contenu dans le vecteur

void trimToSize(): si on a un vecteur avec 2 element dans le vecteur il vas enlever les 8 autres, car elle sont innutile

premier.add ( new Integer ( 5 ) ); ajoute 5 a la premiere liste

premier.size() ?? ( *nombre d'éléments du Vecteur* ) = 1

premier.capacity() ?? ( *capacité totale du Vecteur* ) = 10

deuxieme.add ( new Character ( 'u' )) ; ajoute u a la premiere place

NOUVEAU JDK 1.5 🡪 AUTOBOXING :

deuxieme.add ('t' );

Character lettre = new Character ( 'y' );

deuxieme.insertElementAt ( lettre, 1 );

deuxieme.indexOf ('t'); ?? = 2

deuxieme.size();?? = 3

deuxieme.capacity(); ?? = 4

troisieme.insertElementAt (22.2 , 7 ); ?? = on ne peut pas laisser d espace vide avant l indice 7

deuxieme.remove ( lettre ); ?? =

deuxieme.indexOf ( new Character ('u') ); ?? =

premier.remove ( 4 );?? =

troisieme.add(5.5);

troisième.add(6.5);

troisieme.add(5.5);

troisieme.add(9);

à l'aide d'une boucle for, afficher la somme des éléments présent dans le troisième vecteur :

for (int i = 0; double < troisième.size(); i ++){

somme += troisieme.get(i);

}

Utiliser une boucle for améliorée pour le faire :

Double somme=0;

For (Double d : troisieme)

Somme +=d