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(); Cliquez ou appuyez ici pour entrer du texte.

Vector deuxieme = new Vector(2);Cliquez ou appuyez ici pour entrer du texte.

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

***Méthodes principales :***

void add ( Object o ) :

void insertElementAt ( Object o, int index ) :

Object get ( int index ) :

Object remove ( int index):

boolean remove ( Object o ):

boolean contains (Object o ):

void trimToSize():

EXEMPLES :

Les trois vecteurs :

premier.add ( new Integer ( 5 ) );

premier.size() ?? ( *nombre d'éléments du Vecteur* ) = Cliquez ou appuyez ici pour entrer du texte.

premier.capacity() ?? ( *capacité totale du Vecteur* ) = Cliquez ou appuyez ici pour entrer du texte.

deuxieme.add ( new Character ( 'u' )) ;

NOUVEAU JDK 1.5 🡪 AUTOBOXING :

deuxieme.add ('t' );

Character lettre = new Character ( 'y' );

deuxieme.insertElementAt ( lettre, 1 );

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

deuxieme.size();?? =

deuxieme.capacity(); ?? =

troisieme.insertElementAt (22.2 , 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 :

Utiliser une boucle for améliorée pour le faire :