

1 Vector

1.1 Einleitung

Vektoren sind ähnlich wie ein ArrayList aber ein Vector ist synchronisiert, und enthält viele Methoden die nicht zu den Collection Frameworks gehören. Vektoren sind sehr hilfreich wenn die Größe des Arrays nicht gewusst ist oder wenn ein Array größer werden soll je länger ein Programm läuft.

Die Vector Klasse hat 4 Konstruktoren:

- Vector() Erzeugt ein Vector mit Größe 10.
- Vector(int size) Erzeugt ein Vector mit eingegebene Größe.
- Vector(int size, int incr) Erzeugt ein Vector mit eingegebene Größe und Inkrement, der Inkrement bestimmt wie groß der Vector wird wenn der Vector größer wird.
- Vector(Collection c) Erzeugt ein Vector mit Elementen von den Collection c.

1.2 Methodenbeschreibung

- add()/addAll() - genau wie bei ein ArrayList.
- addElement(Object o) - addiert den Object am ende des Vectors, Vector größe wird um Eins erhöht.
- capacity() - gibt die aktuelle Kapazität des Vectors zurück.
- clear() - löscht alle Elementen in der Vector.
- contains(Object o) - prüft ob ein eingegebene Object in der Vector ist.
- containsAll(Collection c) - prüft ob Alle Elementen in der Collection c in der Vector sind.
- copyInto(Object[] array) - kopiert alle Elementen in der Vector zu ein eingegebene Array.
- elementAt(int index) - gibt Element an der eingegebene Stelle zurück.
- elements() - gibt ein Aufzählung aller Elementen in der Vector zurück
- firstelement() - gibt den erste Element in Vector zurück (index 0)
- get(int index) - gibt den Element an der eingegebene Stelle zurück
- indexOf(Object o) - sucht nach den ersten Treffer von Element in der Vector

- `insertElementAt(Object o, int index)` - addiert Element in eingegebene stelle.
- `isEmpty()` - prüft ob Vector leer ist
- `lastElement()` - gibt den letzten Element in der Vector zurück
- `remove(int index)` - löscht den Element in eingegebene Stelle.
- `removeAll(Collection c)` löscht alle Elementen in Vector die dieser Collection enthält.
- `removeAllElements()` - löscht alle Elementen in den Vector und verändert die größe zu Null.
- `removeElement(Object o)`- löscht erste Treffer von Element in der Vector
- `set(int index, Object o)` - ersetzt den Element von eingegebene Index für ein neues Element
- `setSize()` - verändert die Größe des Vectors
- `size()` - gibt der Größe des Vectors zurück.
- `toArray()` - gibt ein Array mit alle Elementen in Reihenfolge in diesem Vector zurück
- `toString()` - gibt einer String von alle Elementen in den Vector.
- `trimToSize()` - verkleinert die Größe des Vectors zu aktuelle oder benutzte Größe.

1.3 Effizienzanalyse

Graph einfügen und Effizienz ausrechnen. Für einzelne Probleme anwenden.

1.4 Schlussfolgerung

hier die resultierenden Stärken und Schwächen rausschreiben.