Class JavaVector<T>

java.lang.Object
JavaVector<T>

Type Parameters:
`T` - type of the vector

All Implemented Interfaces:
`JavaContainer<T>`

---

public class JavaVector<T> extends Object implements JavaContainer<T>
JavaVector Class implements JavaContainer interface

Field Summary

Fields
Modifier and Type
Field
Description
`private Object[] array`
array of JavaVector
`private int capacity`
capacity of JavaVector
`private int size`
size of JavaVector

Constructor Summary

Constructors
Constructor
Description
`JavaVector(int _capacity)`

constructor creates an empty vector sized 0

- # Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type
Method
Description

`void`
`add(T item)`
adds @param item to the vector

`private void`
`addCap()`
increases capacity by multiplying it with 2 copies the old array to the new one

`boolean`
`equals(JavaVector<T> other)`
overloaded equals method checks if two vectors are equal

`Iterator<T>`
`getIterator()`
returns an iterator for the container

`private boolean`
`isIn(T item)`
checks if the element is in the vector

`void`
`remove(T item)`
removes @param item from the vector

`int`
`size()`
returns the number of elements in the container

`String`
`toString()`

Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

- # Field Details

  - ### capacity

    private int capacity
    capacity of JavaVector

  - ### size

    private int size
    size of JavaVector

  - ### array

    private Object[] array
    array of JavaVector

- # Constructor Details

  - ### JavaVector

    public JavaVector(int _capacity)
    constructor creates an empty vector sized 0

    Parameters:
    `_capacity` - capacity of the vector

- # Method Details

  - ### equals

    public boolean equals(JavaVector<T> other)
    overloaded equals method checks if two vectors are equal

Parameters:
`other` - other vector
Returns:
true if they are equal

add

public void add(T item)
adds @param item to the vector

Specified by:
`add` in interface `JavaContainer<T>`
Parameters:
`item` - element to be added

remove

public void remove(T item)
removes @param item from the vector

Specified by:
`remove` in interface `JavaContainer<T>`
Parameters:
`item` - element to be removed

size

public int size()
Description copied from interface: `JavaContainer`
returns the number of elements in the container

Specified by:
`size` in interface `JavaContainer<T>`
Returns:
size of the vector

getIterator

public Iterator<T> getIterator()
Description copied from interface: `JavaContainer`
returns an iterator for the container

Specified by:
`getIterator` in interface `JavaContainer<T>`
Returns:
iterator of the vector

toString

public String toString()

Overrides:
`toString` in class `Object`
Returns:
string representation of the vector

isIn

private boolean isIn(T item)
checks if the element is in the vector

Parameters:
`item` - element to be checked
Returns:
true if the element is in the vector

addCap

private void addCap()
increases capacity by multiplying it with 2 copies the old array to the new one