- Package
- Class
- Tree
- Index
- Help

SEARCH Search reset

All Classes and Interfaces

All Classes and Interfaces | Interfaces | Classes

Class
Description
JavaContainer<T>
JavaContainer
JavaSet<T>

JavaVector<T>

myIter<T>
myIterator class which implements Iterator interface
test
Driver test code for JavaContainers

- Package
- Class
- Tree
- Index
- Help

SEARCH Search reset

SEARCH reset

All Packages

Package Summary
Package
Description
HOMEWORK

JavaDoc Help

Navigation

Starting from the [Overview](#) page, you can browse the documentation using the links in each page, and in the navigation bar at the top of each page. The [Index](#) and Search box allow you to navigate to specific declarations and summary pages, including: [All Packages](#), [All Classes and Interfaces](#)

Search

You can search for definitions of modules, packages, types, fields, methods, system properties and other terms defined in the API. These items can be searched using part or all of the name, optionally using "camelCase" abbreviations, or multiple search terms separated by whitespace. Some examples:

- `"j.l.obj"` matches "java.lang.Object"
- `"InpStr"` matches "java.io.InputStream"
- `"math exact long"` matches "java.lang.Math.absExact(long)"

Refer to the [Javadoc Search Specification](#) for a full description of search features.

Kinds of Pages

The following sections describe the different kinds of pages in this collection.

Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. These pages may contain the following categories:

- Interfaces
- Classes
- Enum Classes
- Exception Classes
- Annotation Interfaces

Class or Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three

sections consisting of a declaration and description, member summary tables, and detailed member descriptions. Entries in each of these sections are omitted if they are empty or not applicable.

- Class Inheritance Diagram
- Direct Subclasses
- All Known Subinterfaces
- All Known Implementing Classes
- Class or Interface Declaration
- Class or Interface Description

- Nested Class Summary
- Enum Constant Summary
- Field Summary
- Property Summary
- Constructor Summary
- Method Summary
- Required Element Summary
- Optional Element Summary

- Enum Constant Details
- Field Details
- Property Details
- Constructor Details
- Method Details
- Element Details

Note: Annotation interfaces have required and optional elements, but not methods. Only enum classes have enum constants. The components of a record class are displayed as part of the declaration of the record class. Properties are a feature of JavaFX.

The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

Other Files

Packages and modules may contain pages with additional information related to the declarations nearby.

Tree (Class Hierarchy)

There is a [Class Hierarchy](#) page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. Classes are organized by inheritance structure starting with `java.lang.Object`. Interfaces do not inherit from `java.lang.Object`.

- When viewing the Overview page, clicking on TREE displays the hierarchy for all packages.
- When viewing a particular package, class or interface page, clicking on TREE displays the hierarchy for only that package.

All Packages

The [All Packages](#) page contains an alphabetic index of all packages contained in the documentation.

All Classes and Interfaces

The [All Classes and Interfaces](#) page contains an alphabetic index of all classes and interfaces contained in the documentation, including annotation interfaces, enum classes, and record classes.

Index

The [Index](#) contains an alphabetic index of all classes, interfaces, constructors, methods, and fields in the documentation, as well as summary pages such as [All Packages](#), [All Classes and Interfaces](#).

This help file applies to API documentation generated by the standard doclet.

[ ]

Skip navigation links

- [Package](#)
- Class
- [Tree](#)
- Index
- [Help](#)

SEARCH [Search] [reset]

Index

A

[add(T)](#) - Method in interface HOMEWORK.[JavaContainer](#)
  abstract add method which is used to add values to containers
[add(T)](#) - Method in class HOMEWORK.[JavaSet](#)

[add(T)](#) - Method in class HOMEWORK.[JavaVector](#)

E

[equals(Object)](#) - Method in class HOMEWORK.[JavaSet](#)

[equals(Object)](#) - Method in class HOMEWORK.[JavaVector](#)
  equals method checks for two JavaVectors to see if they are equal if user sends same object it returns true if its not
  the same object method checks for two inner arrays of objects first checks their sizes if they are equal checks for
  every element in the array if elements are not same nor in the same order it returns false

G

[getElementAt(int)](#) - Method in class HOMEWORK.[JavaSet](#)
  returns the object at given index, simply an [] indexing operator
[getElementAt(int)](#) - Method in class HOMEWORK.[JavaVector](#)
  simply an indexing operator to return the element in given index
[getIterator()](#) - Method in interface HOMEWORK.[JavaContainer](#)
  abstract method for returning iterators from containers
[getIterator()](#) - Method in class HOMEWORK.[JavaSet](#)

[getIterator()](#) - Method in class HOMEWORK.[JavaVector](#)

H

[hasNext()](#) - Method in class HOMEWORK.[myIter](#)

[HOMEWORK](#) - package HOMEWORK

J

[JavaContainer](#)<[T](#)> - Interface in [HOMEWORK](#)
  JavaContainer
[JavaSet](#)<[T](#)> - Class in [HOMEWORK](#)

[JavaSet()](#) - Constructor for class HOMEWORK.[JavaSet](#)
  we call javaset's parametirezed constructor if the default constructor gets called
[JavaSet(int)](#) - Constructor for class HOMEWORK.[JavaSet](#)
  main constructor of our javaset class
[JavaVector](#)<[T](#)> - Class in [HOMEWORK](#)

[JavaVector()](#) - Constructor for class HOMEWORK.[JavaVector](#)
  default constructor of JavaVector calls parameterized constructor with default value of 20

[JavaVector(int)](#) - Constructor for class HOMEWORK.[JavaVector](#)

 parameterized constructor of our JavaVector class, our innerArray's capacitiy gets assigned to the _c parameter

 this constructor also creates the iterator of our JavaVector

M

[main(String[])](#) - Static method in class HOMEWORK.[test](#)

[myIter](#)<[T](#)> - Class in [HOMEWORK](#)

 myIterator class which implements Iterator interface

[myIter(int, JavaContainer)](#) - Constructor for class HOMEWORK.[myIter](#)

 myIter constructor which is called only from Containers constructors takes 2 parameters

N

[next()](#) - Method in class HOMEWORK.[myIter](#)

 if there is an element in the referenced object's inner array it returns that element

R

[remove(T)](#) - Method in interface HOMEWORK.[JavaContainer](#)

 abstract remove method which is used to remove senden key value from containers

[remove(T)](#) - Method in class HOMEWORK.[JavaSet](#)

[remove(T)](#) - Method in class HOMEWORK.[JavaVector](#)

S

[size()](#) - Method in interface HOMEWORK.[JavaContainer](#)

 abstract size method which is used to return used size in containers

[size()](#) - Method in class HOMEWORK.[JavaSet](#)

[size()](#) - Method in class HOMEWORK.[JavaVector](#)

T

[test](#) - Class in [HOMEWORK](#)

 Driver test code for JavaContainers

[test()](#) - Constructor for class HOMEWORK.[test](#)

[to_txt(Path)](#) - Method in class HOMEWORK.[JavaSet](#)

 to_txt method to write contents of set to an txt file

[to_txt(Path)](#) - Method in class HOMEWORK.[JavaVector](#)

 to_txt method to write contents of vector to an txt file

[toString()](#) - Method in interface HOMEWORK.[JavaContainer](#)

 abstract toString method

[toString()](#) - Method in class HOMEWORK.[JavaSet](#)

 Overriding the toString function for our Java Set class Using mutable StringBuilder class to create our string

[toString()](#) - Method in class HOMEWORK.[JavaVector](#)

 Overriding the toString function for our Java Vector class Using mutable StringBuilder class to create our string

404 Not Found

---

nginx/1.18.0 (Ubuntu)

Skip navigation links

- Package
- Class
- Tree
- Index
- Help

SEARCH [Search] reset

Hierarchy For All Packages

Package Hierarchies:

- HOMEWORK

Class Hierarchy

- java.lang.Object
  - HOMEWORK.JavaSet<T> (implements HOMEWORK.JavaContainer<T>)
  - HOMEWORK.JavaVector<T> (implements HOMEWORK.JavaContainer<T>)
  - HOMEWORK.myIter<T> (implements java.util.Iterator<E>)
  - HOMEWORK.test

Interface Hierarchy

- HOMEWORK.JavaContainer<T>

Search

Search | Reset

▶ Additional resources

The help page provides an introduction to the scope and syntax of JavaDoc search.

You can use the <ctrl> or <cmd> keys in combination with the left and right arrow keys to switch between result tabs in this page.

The URL template below may be used to configure this page as a search engine in browsers that support this feature. It has been tested to work in Google Chrome and Mozilla Firefox. Note that other browsers may not support this feature or require a different URL format.

link | Copy URL

☐ Redirect to first result

Loading search index...