

All Classes

- [JavaContainer](#)
- [JavaContainer.JavaIterator](#)
- [JavaSet](#)
- [JavaVector](#)
- [test2](#)

[Skip navigation links](#)

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

- [All Classes](#)

• SEARCH:

All Classes

- All Classes [Interface Summary](#) [Class Summary](#)

Class	Description
JavaContainer <T>	JavaContainer
JavaContainer.Iterator <T>	Returns the type which is set
JavaSet <T>	A simple implementation of a set using an array.
JavaVector <T>	A simple implementation of a vector using an array.
test2	

[Skip navigation links](#)

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

- [All Classes](#)

[Skip navigation links](#)

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

- [All Classes](#)

SEARCH:

reset

All Packages

- Package Summary

Package **Description**

[homework](#)

[Skip navigation links](#)

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

- [All Classes](#)

[Skip navigation links](#)

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

- [All Classes](#)

SEARCH:

reset

Constant Field Values

Contents

[Skip navigation links](#)

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

- [All Classes](#)

[Skip navigation links](#)

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

- [All Classes](#)

SEARCH:

reset

Deprecated API

Contents

[Skip navigation links](#)

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

- [All Classes](#)

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

- [All Classes](#)

- SEARCH:

How This API Document Is Organized

This API (Application Programming Interface) document has pages corresponding to the items in the navigation bar, described as follows.

• Package

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

- Interfaces
- Classes
- Enums
- Exceptions
- Errors
- Annotation Types

• 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 class/interface description, summary tables, and detailed member descriptions:

- Class Inheritance Diagram
- Direct Subclasses
- All Known Subinterfaces
- All Known Implementing Classes
- Class or Interface Declaration
- Class or Interface Description
- Nested Class Summary
- Field Summary
- Property Summary
- Constructor Summary
- Method Summary
- Field Detail
- Property Detail
- Constructor Detail
- Method Detail

Each summary entry contains the first sentence from the detailed description for that item. 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.

• Annotation Type

Each annotation type has its own separate page with the following sections:

- Annotation Type Declaration
- Annotation Type Description
- Required Element Summary
- Optional Element Summary
- Element Detail

• Enum

Each enum has its own separate page with the following sections:

- Enum Declaration
- Enum Description
- Enum Constant Summary
- Enum Constant Detail

• 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.

• Deprecated API

The [Deprecated API](#) page lists all of the API that have been deprecated. A deprecated API is not recommended for use, generally due to improvements, and a replacement API is usually given. Deprecated APIs may be removed in future implementations.

• Index

The [Index](#) contains an alphabetic index of all classes, interfaces, constructors, methods, and fields, as well as lists of all packages and all classes.

• All Classes

The [All Classes](#) link shows all classes and interfaces except non-static nested types.

• Serialized Form

Each serializable or externalizable class has a description of its serialization fields and methods. This information is of interest to re-implementors, not to developers using the API. While there is no link in the navigation bar, you can get to this information by going to any serialized class and clicking "Serialized Form" in the "See also" section of the class description.

• Constant Field Values

The [Constant Field Values](#) page lists the static final fields and their values.

• Search

You can search for definitions of modules, packages, types, fields, methods and other terms defined in the API, using some or all of the name. "Camel-case" abbreviations are supported: for example, "InpStr" will find "InputStream" and "InputStreamReader".

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

[Skip navigation links](#)

- [Package](#)
- [Class](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)
- [All Classes](#)

404 Not Found

nginx/1.18.0 (Ubuntu)

[Skip navigation links](#)

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

- [All Classes](#)

• SEARCH:

[A](#) [E](#) [G](#) [H](#) [I](#) [J](#) [M](#) [N](#) [R](#) [S](#) [T](#)
[All Classes](#) [All Packages](#)

A

[add\(T\)](#) - Method in interface homework.[JavaContainer](#)

Adds an element to the container.

[add\(T\)](#) - Method in class homework.[JavaSet](#)

Adds the specified element to the set if it is not already present.

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

Adds the specified element to the vector.

E

[equals\(Object\)](#) - Method in interface homework.[JavaContainer](#)

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

Checks if this set is equal to another object.

G

[getIterator\(\)](#) - Method in class homework.[JavaSet](#)

Returns an iterator over the elements in the set.

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

Returns an iterator over the elements in the vector.

H

[hasNext\(\)](#) - Method in interface homework.[JavaContainer](#).[JavaIterator](#)

Checks if there is a next element in the iterator.

[hasNext\(\)](#) - Method in class homework.[JavaVector](#).[VectorIterator](#)

Checks if there is a next element in the iterator.

[homework](#) - package homework

I

[is_included\(Object\)](#) - Method in class homework.[JavaSet](#)

Checks if the specified element is included in the set.

J

[JavaContainer<T>](#) - Interface in [homework](#)

[JavaContainer](#)

[JavaContainer](#).[JavaIterator<T>](#) - Interface in [homework](#)

Returns the type which is set

[JavaSet<T>](#) - Class in [homework](#)

A simple implementation of a set using an array.

[JavaSet\(\)](#) - Constructor for class homework.[JavaSet](#)

Constructs an empty set with an initial capacity of 10.

[JavaVector<T>](#) - Class in [homework](#)

A simple implementation of a vector using an array.

[JavaVector\(\)](#) - Constructor for class homework.[JavaVector](#)

Constructs an empty vector with an initial capacity of 10.

[JavaVector](#).[VectorIterator](#) - Class in [homework](#)

An iterator implementation for iterating over elements in the vector.

M

[main\(String\[\]\)](#) - Static method in class homework.[test2](#)

N

[next\(\)](#) - Method in interface homework.[JavaContainer.Iterator](#)

Returns the next element in the iterator and advances the iterator.

[next\(\)](#) - Method in class homework.[JavaVector.VectorIterator](#)

Returns the next element in the iterator and advances the iterator.

R

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

Removes the specified element from the container.

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

Removes the specified element from the set.

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

Removes the specified element from the vector.

S

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

Returns the number of elements in the container.

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

Returns the number of elements in the set.

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

Returns the number of elements in the vector.

T

[test2](#) - Class in [homework](#)

[test2\(\)](#) - Constructor for class homework.[test2](#)

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

Returns a string representation of the set.

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

Returns a string representation of the vector.

[A](#) [E](#) [G](#) [H](#) [I](#) [J](#) [M](#) [N](#) [R](#) [S](#) [T](#)

[All Classes](#) [All Packages](#)

[Skip navigation links](#)

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

- [All Classes](#)

[Skip navigation links](#)

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

- [All Classes](#)

- SEARCH:

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.[JavaVector.VectorIterator](#) (implements homework.[JavaContainer.Javalterator](#)<T>)
 - homework.[test2](#)

Interface Hierarchy

- homework.[JavaContainer](#)<T>
- homework.[JavaContainer.Javalterator](#)<T>

[Skip navigation links](#)

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

- [All Classes](#)