

Class BST12Adapt<E extends java.lang.Comparable<? super E>>

java.lang.Object
BST12Adapt<E>

All Implemented Interfaces:

BinSearchTree12<E>

```
public class BST12Adapt<E extends java.lang.Comparable<? super E>>
extends java.lang.Object
implements BinSearchTree12<E>
```

BST12Adapt is the adapted class that adapts the TreeSet Class from the java collections framework. It implements BinSearchTree12, which is defined as a subset of the methods defined in the Java Collections Framework class Treetset, with 2 additional methods: height() and numChildren().

Nested Class Summary

Nested Classes	
Modifier and Type	Class and Description
protected class	BST12Adapt.BST12AdaptIterator INNER CLASS

Constructor Summary

Constructors	
Constructor and Description	
BST12Adapt ()	no arg constructor constructs a new, empty BST12Adapt object, sorted according to the natrual ordering of its elements.
BST12Adapt (java.util.Collection<? extends E > c)	Constructor.

Method Summary

Modifier and Type	Method and Description
boolean	add (E e) Adds the specified element to this set if ti is not already present.
boolean	addAll (java.util.Collection<? extends E > c) Adds all of the elements in the specified collection to this set.
void	clear () removes all of the elements from this search tree.
boolean	contains (E o) Returns true if this tree contains the specified element.
E	first () returns the first (lowest) element currently in this set.
int	height () Additional methods defined in BinSearchTree12 interface.
boolean	isEmpty () returns true if this set contains no element
java.util.Iterator< E >	iterator () Returns an iterator over the elements in this set in ascending order.
E	last () returns the last (highest) element currently in this set.
int	numChildren (E target) Additional methods defined in BinSearchTree12 interface.
boolean	remove (E o) removes the specified element from this search tree if it is present
int	size () returns the number of elements in the tree
java.lang.String	toString () toString method redefined for tester file usage.

Methods inherited from class java.lang.Object

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

Constructor Detail

BST12Adapt

```
public BST12Adapt()
```

no arg constructor constructs a new, empty BST12Adapt object, sorted according to the natural ordering of its elements.

BST12Adapt

```
public BST12Adapt(java.util.Collection<? extends E> c)
```

Constructor. Constructs a new BST12Adapt object, containing the elements in the specified collection, sorted according to the natural ordering of its elements.

Method Detail

add

```
public boolean add(E e)
```

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

Specified by:

add in interface `BinSearchTree12<E extends java.lang.Comparable<? super E>>`

Parameters:

e - element to be added to this set

Returns:

true if this set did not already contain the specified element.

Throws:

`java.lang.ClassCastException` - if the specified object cannot be compared with the elements currently in this set.

`java.lang.NullPointerException` - if the specified element is null and this set does not permit null elements.

addAll

```
public boolean addAll(java.util.Collection<? extends E> c)
```

Adds all of the elements in the specified collection to this set.

Specified by:

addAll in interface `BinSearchTree12<E extends java.lang.Comparable<? super E>>`

Parameters:

c - collection containing elements to be added to this set.

Returns:

true if this set changed as a result of the call

Throws:

`java.lang.ClassCastException` - if the specified object cannot be compared with the elemtns currently in this set.

`java.lang.NullPointerException` - if the specified element is null and this sets does not permit null elements.

clear

```
public void clear()
```

removes all of the elements from this search tree.

Specified by:

clear in interface `BinSearchTree12<E extends java.lang.Comparable<? super E>>`

contains

```
public boolean contains(E o)
```

Returns true if this tree contains the specified element.

Specified by:

contains in interface `BinSearchTree12<E extends java.lang.Comparable<? super E>>`

Parameters:

o - object to be checked for containment in this set.

Returns:

true if this set contains the specified element

Throws:

`java.lang.ClassCastException` - if the specified object cannot be compared with the elemtns currently in this set.

`java.lang.NullPointerException` - if the specified element is null and this sets does not permit null elements.

first

```
public E first()
```

returns the first (lowest) element currently in this set.

Specified by:

first in interface `BinSearchTree12<E extends java.lang.Comparable<? super`

`E>>`

Returns:

the lowest element in this set

Throws:

`java.util.NoSuchElementException` - if this set is empty

last

```
public E last()
```

returns the last (highest) element currently in this set.

Specified by:

`last` in interface `BinSearchTree12<E extends java.lang.Comparable<? super E>>`

Returns:

the highest element in this set

Throws:

`java.util.NoSuchElementException` - if this set is empty

isEmpty

```
public boolean isEmpty()
```

returns true if this set contains no element

Specified by:

`isEmpty` in interface `BinSearchTree12<E extends java.lang.Comparable<? super E>>`

Returns:

true if this set contains no element

iterator

```
public java.util.Iterator<E> iterator()
```

Returns an iterator over the elements in this set in ascending order.

Specified by:

`iterator` in interface `BinSearchTree12<E extends java.lang.Comparable<? super E>>`

remove

```
public boolean remove(E o)
```

removes the specified element from this search tree if it is present

Specified by:

remove in interface BinSearchTree12<**E** extends java.lang.Comparable<? super **E**>>

Parameters:

o - object to be removed from this set, if present

Returns:

true if this set contained the specified element

Throws:

java.lang.ClassCastException - if the specified object cannot be compared with the elemtns currently in this set.

java.lang.NullPointerException - if the specified element is null and this sets does not permit null elements.

size

```
public int size()
```

returns the number of elements in the tree

Specified by:

size in interface BinSearchTree12<**E** extends java.lang.Comparable<? super **E**>>

height

```
public int height()
```

Additional methods defined in BinSearchTree12 interface. returns the height of the tree. An empty tree returns 0. 1 for a tree with one node. size() for all other cases.

Specified by:

height in interface BinSearchTree12<**E** extends java.lang.Comparable<? super **E**>>

numChildren

```
public int numChildren(E target)
```

Additional methods defined in BinSearchTree12 interface. Return -1 if the target node is in the tree.

Specified by:

numChildren in interface `BinSearchTree12<E extends java.lang.Comparable<? super E>>`

Throws:

`java.util.NoSuchElementException` – if the target node is not in the tree

`java.lang.IllegalArgumentException` – for any other faults.

toString

```
public java.lang.String toString()
```

toString method redefined for tester file usage.

Overrides:

toString in class `java.lang.Object`

Returns:

a content of the tree using In-Order traversal.

[PACKAGE](#) [CLASS](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)

[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)