

Class BST12<E extends java.lang.Comparable<? super E>>

java.lang.Object
BST12<E>

All Implemented Interfaces:

BinSearchTree12<E>

```
public class BST12<E extends java.lang.Comparable<? super E>>
extends java.lang.Object
implements BinSearchTree12<E>
```

BST12 implements BinSearchTree12, which is defined as a subset of the methods defined in the Java Collections Framework class TreeSet, with 2 additional methods: height() and numChildren().

Nested Class Summary

Nested Classes	
Modifier and Type	Class and Description
protected class	BST12.BST12Iterator INNER CLASS for an Iterator
protected class	BST12.BST12Node < E extends java.lang.Comparable<? super E >> INNER CLASS for a Node A BST is composed of BST12Nodes.

Constructor Summary

Constructors	
Constructor and Description	
BST12 ()	no-arg constructor constructs a new, empty BST12Adapt object, sorted according to the natural ordering of its elements.
BST12 (java.util.Collection<? extends E > c)	constructor that constructs a new BST containing the elements in the specified collection.

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type		Method and Description
protected	BST12.BST12Node<E>	add (BST12.BST12Node<E> parent, BST12.BST12Node<E> curr, E e) protected version that does the work for add().
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.
protected boolean		contains (BST12.BST12Node<E> node, E o) Protected version that does the work for contains(E o) Has direct access to node's data fields.
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 that returns the height of the tree.
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) Returns the number of children of the Node that references target.
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

BST12

```
public BST12()
```

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

BST12

```
public BST12(java.util.Collection<? extends E> c)
```

constructor that constructs a new `BST` containing the elements in the specified collection.

Parameters:

`c` - the collection to be added

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.

add

```
protected BST12.BST12Node<E> add(BST12.BST12Node<E> parent,  
                                BST12.BST12Node<E> curr,  
                                E e)
```

protected version that does the work for `add()`. Using recursion, this helper method returns a reference to the current node to the caller in base case.

Parameters:

`parent` – the parent of the current `BST12Node`

`curr` – the current `BST12Node` in the recursive call

`e` – the element to be added

Returns:

a reference to the current node to the caller

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 elements correctly in this set

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

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.

contains

```
protected boolean contains(BST12.BST12Node<E> node,  
                           E o)
```

Protected version that does the work for `contains(E o)` Has direct access to node's data fields. This helper method uses recursion to perform comparison, stops at base case where there is no more child.

Parameters:

`node` - a `BST12Node` to check element containment

`o` - element to be checked

Returns:

true if found containment; false otherwise

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

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 elements currently in this set.

`java.lang.NullPointerException` - if the specified element is null and this set 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>>
```

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 that returns the height of the tree. An empty tree returns 0, a tree with one ele returns a height of 1.

Specified by:

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

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

numChildren

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

Returns the number of children of the Node that references target.

Specified by:

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

Throws:

`java.lang.IllegalArgumentException` - if null pointer or if there's a `ClassCastException` (any other problems)

`java.util.NoSuchElementException` - if target is not found in the tree

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](#)