Package Tree

Class BST<K extends java.lang.Comparable<K>>

java.lang.Object
    Tree.BST<K>

All Implemented Interfaces:

`java.lang.Iterable<K>`

```
public class BST<K extends java.lang.Comparable<K>>
extends java.lang.Object
implements java.lang.Iterable<K>
```

Since:

2021. DETAILS..... a generic (any datatype K - like another comparable data structure like an array) Binary Search Tree for searching, computational data storage and an add-on of collecting the data if needed into a list. It's collection is iterable. https://www.geeksforgeeks.org/binary-search-tree-set-2-delete/

Version:

1.0

Author:

David Nathaniel (Ambassador of Christ)

Constructor Summary

Constructors

Constructor	Description
`BST()`	Default Constructor.
`BST(Tree.BST.Node root)`	A copy constructor.

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`java.util.ArrayList<K>`	`collection()`	Collects the data in the BST.

BST<K>	insert(K data)	Insert a data into the BinarySearchTree.
java.util.Iterator<K>	iterator()	Returns an iterator over the collection of BST's data.
static void	main (java.lang.String[] args)	
void	postOrder()	Basic postOrder traversal of a BST prints them left, right subtree, then the root
void	preOrder()	Basic preOrder traversal of a BST prints them root, left, and right subtree.
boolean	search(K data)	search if a data is in the BST.
void	selectOrder (java.lang.String in)	Select the order that you want the BST to be traversed.
int	size()	Size of the BST

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface java.lang.Iterable

forEach, spliterator

Constructor Detail

BST

public BST()

Default Constructor. Root is empty.

collection of data.

Parameters:

`root` - -- root of the BST

Method Detail

insert

```
public BST<K> insert(K data)
```

Insert a data into the BinarySearchTree. Pass in old root to add new data into it. Now reassign root to its modified form

Parameters:

`data` - -- generic (any datatype) in the BST

Returns:

a BST

search

```
public boolean search(K data)
```

search if a data is in the BST. returns true if present or false otherwise.

Parameters:

`data` - -- generic (any datatype) in the BST

Returns:

true or false

delete

```
public BST<K> delete(K data)
```

a BST

size

```
public int size()
```

Size of the BST

Returns:

integer (the number of nodes in the BST).

collection

```
public java.util.ArrayList<K> collection()
```

Collects the data in the BST. Uses a recursive method to get all data.

Returns:

an ArrayList of data in BST.

inOrder

```
public void inOrder()
```

Basic inOrder traversal of a BST prints them in the natural order. prints them left, root, and right subtree.

preOrder

```
public void preOrder()
```

Basic preOrder traversal of a BST prints them root, left, and right subtree.

postOrder

selectOrder

```
public void selectOrder(java.lang.String in)
```

Select the order that you want the BST to be traversed.

Parameters:

`in` - -- (input string) "in" - for inOrder traversal. "pre" - for preOrder traversal. Else a postOrder traversal.

main

```
public static void main(java.lang.String[] args)
```

iterator

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

Returns an iterator over the collection of BST's data. hasNext() - If there is more data from the data collection, then return true. Else false. next() - returns the data.

Specified by:

`iterator` in interface `java.lang.Iterable<K extends java.lang.Comparable<K>>`

Throws:

`java.util.NoSuchElementException` - -- cannot remove a nonexisting data.