PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV CLASS **NEXT CLASS** FRAMES NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

p7_package

# Class Generic_BST_Class<GenericData extends java.lang.Comparable<GenericData>>

java.lang.Object
    p7_package.Generic_BST_Class<GenericData>

**Type Parameters:**

`GenericData -`

```
public class Generic_BST_Class<GenericData extends java.lang.Comparable<GenericData>>
extends java.lang.Object
```

Binary Sesarch Tree (BST) class for managing generic data

Note: Data used must have implemented Comparable interface

**Author:**

`MichaelL`

## *Nested Class Summary*

### Nested Classes

| Modifier and Type | Class and Description |
|---|---|
| `private class` | `Generic_BST_Class.BST_Node`<br>Binary Search Tree node class for managing generic data |

## *Field Summary*

### Fields

| Modifier and Type | Field and Description |
|---|---|
| `private Generic_BST_Class.BST_Node` | `BST_Root`<br>Root of BST |

## *Constructor Summary*

### Constructors

| Constructor and Description |
|---|
| `Generic_BST_Class()`<br>Default class constructor, initializes BST |

## *Method Summary*

**All Methods** **Instance Methods** **Concrete Methods**

| Modifier and Type | Method and Description |
|---|---|
| `void` | `clearTree()`<br>Clears tree |
| `void` | `displayInOrder()`<br>Provides inOrder traversal action |
| `private void` | `displayInOrderHelper(Generic_BST_Class.BST_Node localRoot)`<br>Provides inOrder traversal action using recursion |
| `void` | `displayPostOrder()`<br>Provides postOrder traversal action |
| `private void` | `displayPostOrderHelper(Generic_BST_Class.BST_Node localRoot)`<br>Provides postOrder traversal action using recursion |
| `void` | `displayPreOrder()`<br>Provides preOrder traversal action |
| `private void` | `displayPreOrderHelper(Generic_BST_Class.BST_Node localRoot)`<br>Provides preOrder traversal action using recursion |
| `void` | `insert(GenericData inData)`<br>Insert method for BST |
| `private Generic_BST_Class.BST_Node` | `insertHelper(Generic_BST_Class.BST_Node localRoot, GenericData inData)`<br>Insert helper method for BST insert action |
| `boolean` | `isEmpty()`<br>Test for empty tree |
| `private Generic_BST_Class.BST_Node` | `removeFromMax(Generic_BST_Class.BST_Node maxParent, Generic_BST_Class.BST_Node maxLoc)`<br>Searches tree from given node to maximum value node below it, stores data value found, and then unlinks the node |
| `GenericData` | `removeItem(GenericData inData)`<br>Removes data node from tree using given key |
| `private Generic_BST_Class.BST_Node` | `removeItemHelper(Generic_BST_Class.BST_Node localRoot, GenericData outData)`<br>Remove helper for BST remove action |
| `GenericData` | `search(GenericData searchData)`<br>Searches for data in BST given GenericData with necessary key |
| `private GenericData` | `searchHelper(Generic_BST_Class.BST_Node localRoot, GenericData searchData)`<br>Helper method for BST search action |

### Methods inherited from class java.lang.Object

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

## *Field Detail*

### BST_Root

```
private Generic_BST_Class.BST_Node BST_Root
```

Root of BST

## *Constructor Detail*

### Generic_BST_Class

```
public Generic_BST_Class()
```

Default class constructor, initializes BST

## *Method Detail*

### clearTree

```
public void clearTree()
```

Clears tree

### displayInOrder

```
public void displayInOrder()
```

Provides inOrder traversal action

Note: Calls displayInOrderHelper

### displayInOrderHelper

```
private void displayInOrderHelper(Generic_BST_Class.BST_Node localRoot)
```

Provides inOrder traversal action using recursion

**Parameters:**

`localRoot - BST_Node tree root reference at the current recursion level`

### displayPostOrder

```
public void displayPostOrder()
```

Provides postOrder traversal action

Note: Calls displayPostOrderHelper

### displayPostOrderHelper

```
private void displayPostOrderHelper(Generic_BST_Class.BST_Node localRoot)
```

Provides postOrder traversal action using recursion

**Parameters:**

`localRoot - BST_Node tree root reference at the current recursion level`

### displayPreOrder

```
public void displayPreOrder()
```

Provides preOrder traversal action

Note: Calls displayPreOrderHelper

### displayPreOrderHelper

```
private void displayPreOrderHelper(Generic_BST_Class.BST_Node localRoot)
```

Provides preOrder traversal action using recursion

**Parameters:**

`localRoot - BST_Node tree root reference at the current recursion level`

### insert

```
public void insert(GenericData inData)
```

Insert method for BST

Note: uses insert helper method which returns root reference

Note: uses search to verify that key is not already in tree; if key is already in tree, insert is not conducted

**Parameters:**

`inData - GenericData data to be added to BST`

### insertHelper

```
private Generic_BST_Class.BST_Node insertHelper(Generic_BST_Class.BST_Node localRoot,
                                                GenericData inData)
```

Insert helper method for BST insert action

Note: Recursive method returns updated local root to maintain tree linkage

**Parameters:**

`localRoot - BST_Node tree root reference at the current recursion level`

`inData - GenericData item to be added to BST`

**Returns:**

`BST_Node reference used to maintain tree linkage`

### isEmpty

```
public boolean isEmpty()
```

Test for empty tree

**Returns:**

`Boolean result of test`

### removeFromMax

```
private Generic_BST_Class.BST_Node removeFromMax(Generic_BST_Class.BST_Node maxParent,
                                                 Generic_BST_Class.BST_Node maxLoc)
```

Searches tree from given node to maximum value node below it, stores data value found, and then unlinks the node

**Parameters:**

`maxParent - BST_Node reference to current node`

`maxLoc - BST_Node reference to child node to be tested`

**Returns:**

`BST_Node reference containing removed node`

### removeItem

```
public GenericData removeItem(GenericData inData)
```

Removes data node from tree using given key

Note: uses remove helper method

Note: uses search initially to get value, if it is in tree; if value found, remove helper method is called, otherwise returns null

**Parameters:**

`inData - GenericData that includes the necessary key`

**Returns:**

`GenericData result of remove action`

### removeItemHelper

```
private Generic_BST_Class.BST_Node removeItemHelper(Generic_BST_Class.BST_Node localRoot,
                                                    GenericData outData)
```

Remove helper for BST remove action

Note: Recursive method returns updated local root to maintain tree linkage

Note: uses removeFromMax method

**Parameters:**

`localRoot - BST_Node tree root reference at the current recursion level`

`outData - GenericData item that includes the necessary key`

**Returns:**

`BST_Node reference result of remove helper action`

### search

```
public GenericData search(GenericData searchData)
```

Searches for data in BST given GenericData with necessary key

**Parameters:**

`searchData - GenericData item containing key`

**Returns:**

`GenericData reference to found data`

### searchHelper

```
private GenericData searchHelper(Generic_BST_Class.BST_Node localRoot,
                                 GenericData searchData)
```

Helper method for BST search action

**Parameters:**

`localRoot - BST_Node tree root reference at the current recursion level`

`searchData - GenericData item containing key`

**Returns:**

`GenericData item found`

PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV CLASS **NEXT CLASS** FRAMES NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD