



Package [main](#)

## Class InternalNode

[java.lang.Object](#)  
  [main.BaseNode](#)  
    [main.InternalNode](#)

```
public class InternalNode  
extends BaseNode
```

The InternalNode class models an InternalNode of our QuadTree with 4 children

Author:

calchen

### Constructor Summary

#### Constructors

Constructor	Description
<a href="#">InternalNode()</a>	Empty constructor of the InternalNode class, which initializes all 4 children to EmptyNode
<a href="#">InternalNode(<a href="#">BaseNode</a> northE, <a href="#">BaseNode</a> northW, <a href="#">BaseNode</a> southE, <a href="#">BaseNode</a> southW)</a>	Copy constructor of the InternalNode class, which initializes all 4 children to given BaseNodes
<a href="#">InternalNode(<a href="#">Range</a> range)</a>	Constructor of the InternalNode class, set the range of this node

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type	Method	Description
<a href="#">BaseNode</a>	<a href="#">getNorthE()</a>	Getter for the north-east child of this node
<a href="#">BaseNode</a>	<a href="#">getNorthW()</a>	Getter for the north-west child of this node
<a href="#">BaseNode</a>	<a href="#">getSouthE()</a>	Getter for the south-east child of this node



void	<b>search</b> (java.lang.String type, <b>Range</b> range, java.util.List< <b>Location</b> > locs)	search() searches locations of given type within given range in children node representing four directions NE, NW, SE, SW
void	<b>setNorthE</b> ( <b>BaseNode</b> northE)	Setter for the north-east child of this node
void	<b>setNorthW</b> ( <b>BaseNode</b> northW)	Setter for the north-west child of this node
void	<b>setSouthE</b> ( <b>BaseNode</b> southE)	Setter for the south-east child of this node
void	<b>setSouthW</b> ( <b>BaseNode</b> southW)	Setter for the south-west child of this node

**Methods inherited from class main.BaseNode**`getRange, mathSplit, setRange`**Methods inherited from class java.lang.Object**`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`**Constructor Detail****InternalNode**`public InternalNode()`

Empty constructor of the InternalNode class, which initializes all 4 children to EmptyNode

**InternalNode**`public InternalNode(Range range)`

Constructor of the InternalNode class, set the range of this node

**Parameters:**

range -

**InternalNode**



Copy constructor of the InternalNode class, which initializes all 4 children to given BaseNodes

## Method Detail

### search

```
public void search(java.lang.String type,
                  Range range,
                  java.util.List<Location> locs)
```

search() searches locations of given type within given range in children node representing four directions NE, NW, SE, SW

**Specified by:**

[search](#) in class [BaseNode](#)

**Parameters:**

type - the type of Location we want to search

range - the search Range

locs - a list of Locations containing the search results (modify in-place)

### isEmpty

```
public boolean isEmpty()
```

isEmpty() checks if all 4 children of this InternalNode are EmptyNodes

**Specified by:**

[isEmpty](#) in class [BaseNode](#)

**Returns:**

true if all 4 children of this InternalNode are EmptyNodes and false otherwise

### getNorthE

```
public BaseNode getNorthE()
```

Getter for the north-east child of this node

**Returns:**

the north-east child of this node



```
public void setNorthE(BaseNode northE)
```

Setter for the north-east child of this node

**Parameters:**

northE - the new north-east child of this class

**getNorthW**

```
public BaseNode getNorthW()
```

Getter for the north-west child of this node

**Returns:**

the north-west child of this node

**setNorthW**

```
public void setNorthW(BaseNode northW)
```

Setter for the north-west child of this node

**Parameters:**

northW - the new north-west node of this node

**getSouthE**

```
public BaseNode getSouthE()
```

Getter for the south-east child of this node

**Returns:**

the south-east child of this node

**setSouthE**

```
public void setSouthE(BaseNode southE)
```

Setter for the south-east child of this node

**Parameters:**

```
public BaseNode getSouthW()
```

Getter for the south-west child of this node

**Returns:**

the south-west node of this node

### setSouthW

```
public void setSouthW(BaseNode southW)
```

Setter for the south-west child of this node

**Parameters:**

southW - the new south-west child of this node